Tassa pagata per campione allegato

Anno XVIII - N. 7 - Settembre 1989 - L. 8.500

**SUONI**
**EFFETTI SPECIALI FATTI IN CASA**

**ACCESSORI**
**ACTION REPLAY MK5 E IL JOYSTICK A SENSORI**

**UTILITY**
**CAMBIA LOOK AI TUOI PROGRAMMI**

**PROGRAMMAZIONE**
**ANCORA QPL, LOGO, ULTRA BASIC E TIPS**

Sped. in Abb. Postale Gr. III/70%

**EQUO CANONE 2**
**Contratto automatico a norma di legge**

**Trasferimento automatico dei programmi da cassetta a disco**

RED HEAT
- DANKO -
L'Est e l'Ovest mettono insieme le loro forze per combattere un mercante di droga sovietico. Due detective - uno russo e uno di Chicago - sono spinti da motivi diversi, ma la preda é una sola.
- DANKO -
Il più caldo "Tie - in" del momento, tutto azione e grafica superba.

Radio Elettronica & Computer

N. 7 - Settembre 1989

*Rubriche:*

# Sommario



## Caricate così i programmi della cassetta allegata:

*Riavvolgete il nastro e digitate **LOAD** seguito da **RETURN** sulla tastiera del C64 e **PLAY** sul registratore. Verrà caricato il programma di presentazione con il menù dei programmi. Digitate **RUN** seguito dalla pressione del tasto **RETURN**. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare: **LOAD "NOME PROGRAMMA"** seguito dalla pressione del tasto **RETURN**.*


***Direttore Editoriale***
***Area Informatica***
Marinella Zetti

***Direttore responsabile***
Paolo Romani

***Caporedattore***
Fernando Zanini

***Responsabile grafico***
***Desktop Publishing***
Adelio Barcella

***Impaginazione elettronica***
Denise De Matteis

***Segretaria di redazione***
Alessandra Marini

***Collaboratori***
Paolo Gussoni, Isa Sestini





*RadioELETTRONICA&COMPUTER*
Rivista mensile, una copia L. 8.500, numeri arretrati lire 10.000 cadauno.
Pubblicazione mensile registrata presso il Tribunale di Monza n. 679 del 28/11/88.

***Fotolito:*** Bassoli - Milano.
***Stampa:*** GEMM Grafica srl, Paderno Dugnano (MI).
***Diffusione:*** Concessionario esclusivo per l'Italia A.&G. Marco SpA, Via Fortezza 27 - 20126 Milano.
Spedizione in abb. post. gruppo III/70.

***Abbonamenti:*** Annuale L. 54.000, estero L. 83.000. Biennale L. 97.500

RadioELETTRONICA & COMPUTER è titolare in esclusiva per l'Italia dei testi e dei progetti di Radio Plans e Electronique Pratique, periodici del gruppo Societé Parisienne d'Edition.

**Gruppo Editoriale JCE srl**
*Sede legale, Direzione, Redazione, Amministrazione*
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/66025.1 -Telex 352376
JCE MIL I - Telefax 61.27.620

***Direzione Amministrativa:*** Walter Buzzavo

***Direttore Commerciale:*** Giuseppe Tiani

***Pubblicità e Marketing***
Gruppo Editoriale JCE - Divisione Pubblicità
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/66025.1

***Concessionario eslusivo per Roma, Lazio e centro sud:***
UNION MEDIA srl - Via C. Fracassini, 18
00198 Roma - Tel. 06/3215434 (13 linee R.A.)
Telex 630206 UNION I - Telefax 06/3215678

***Abbonamenti:*** Le richieste di informazioni sugli abbonamenti in corso si ricevono per telefono tutti i giorni lavorativi dalle ore 9 alle 12. Tel. 02/66025311- 66025338

***Spedizioni:*** Daniela Radicchi

I versamenti vanno indirizzati a:
Gruppo Editoriale JCE srl, Via Ferri 6
20092 Cinisello Balsamo (MI), mediante l'emissione di assegno circolare, cartolina vaglia o utilizzando il c.c.p. n. 351205. Per i cambi di indirizzo allegare alla comunicazione l'importo di L. 3.000, anche in francobolli, e indicare insieme al nuovo anche il vecchio indirizzo.

Associato al

Consorzio Stampa Specializzata Tecnica

Testata in corso di certificazione obbligatoria secondo quanto stabilito dal Regolamento del C.S.S.T.

Mensile associato all'USPI Unione Stampa Periodica italiana

Questo mese accantoniamo i programmi seri per dedicarci esclusivamente ai videogiochi.

### Lords Of The Rising Sun

I "Signori del sole nascente", questo è il nome del videogioco, è un nuovo eccellente prodotto della Cinemaware, ovvero il film fatto software. Lo slogan della software house è "The home of interactive movies", ossia la casa dei film interattivi. I primi acquirenti di videogiochi Amiga conoscono senz'altro il bellissimo Defender of The Crown, che è il primo prodotto Cinemaware che ha visto la luce nel ricchissimo mercato dei videogiochi per Amiga. Se invece non conoscete i prodotti Cinemaware e non sapete che cos'è un "film interattivo" (forse la definizione è un po' pomposa) vi diciamo che si tratta di un programma che unisce le caratteristiche degli adventure con l'animazione: possiamo dire che sono adventure animati, o se preferite animazioni in cui voi intervenite con le vostre scelte. Lords Of The Rising Sun è il degno erede di questo tipo di videogiochi ed è a nostro avviso uno dei migliori giochi prodotti per Amiga, anche se non è tutto oro quel che luccica.

Avete senz'altro capito che i signori del sole nascente sono i potenti del Giappone, infatti l'azione si svolge in tale paese nel dodicesimo secolo. In tale periodo il paese è sconvolto da una guerra civile e le potenti famiglie dei Taira e dei Minamoto combattono accanitamenteper la conquista del trono. Voi controllate i Minamoto e potete scegliere di giocare come politico o come comandante militare. Nel primo caso assumete il ruolo di Yorimoto, "l'Andreotti" del Giappone medievale, e nel secondo caso indossate le vesti del fratello Yoshitsune, grande guerriero ed eccezionale stratega militare, il "Patton" della situazione.

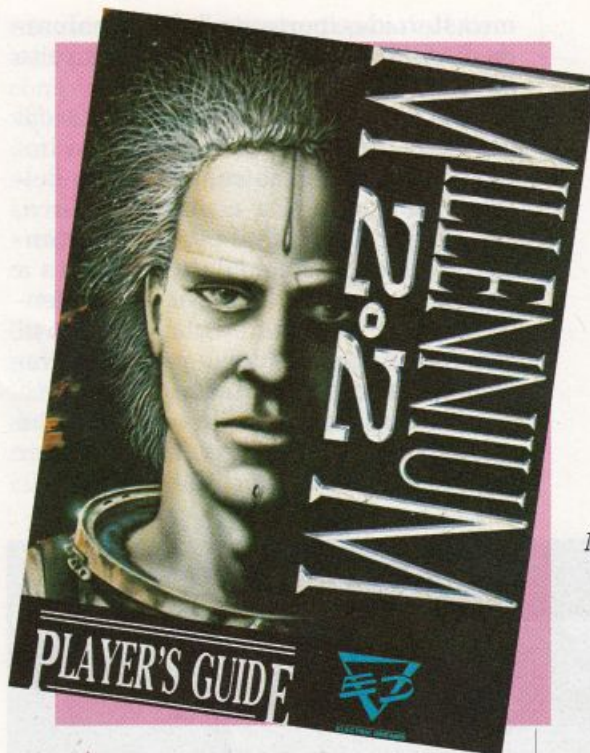
La situazione di gioco è la seguente: i Taira hanno riportato diverse vittorie, hanno assassinato il padre di Yorimoto e Yoshitsune e hanno spodestato l'imperatore. Vi trovate dunque in una posizione molto difficile in quanto partite in svantaggio e vi giocate tutto quel che possedete. Il vostro compito è molto delicato poiché dovete comandare imponenti armate di guerrieri Samurai, dovete difendere la vostra vita contro i ripetuti attacchi dei temibili Ninja e soprattutto dovete giocare abilmente tutte le vostre armi diplomatiche stringendo alleanze con le altre famiglie. Se saprete giocare bene le vostre carte diventerete Shogun, ovvero un comandante forte, temuto e onorato. Naturalmente in caso di insuccesso non potrete sfuggire alla tradizione giapponese: l'eventuale vittoria dei Taira coinciderà con la caduta in disgrazia della vostra famiglia e con l'ineluttabile Kara-Kiri. Vi abbiamo descritto dettagliatamente la situazione di gioco perché riteniamo che Lords Of The Rising Sun sia uno dei più bei giochi creati per Amiga. Aggiungiamo che la dinamica del gioco, molto appassionante, è supportata da un'eccellente grafica e anche da ottimi effetti sonori. Il gioco si controlla interamente con il mouse. Pur essendo molto positivo il nostro giudizio su questo prodotto Cinemaware, non possiamo nascondervi un paio di difetti. Il primo consiste nel fatto che il gioco è molto complesso: non si tratta certamente di un videogioco rilassante, al contrario richiede molto tempo di studio per essere affrontato con successo. Questo non costituisce certamente un problema per gli appassionati di adventure, che solitamente non si divertono se non passano le notti insonni per capire come si affronta una certa situazione, tuttavia

*Vindicators. La schermata di presentazione del gioco. In primo piano il carro armato strategico SR-88.*

può scoraggiare rapidamente chi considera i videogiochi come un rilassante divertimento. L'altro difetto, purtroppo ricorrente in questo tipo di programmi, è la mancata traduzione del programma; esso assume una notevole importanza poiché la partita è sovente interrotta da messaggi in inglese (non in giapponese antico, non preoccupatevi!) che comunicano l'evolversi della situazione, e non è certamente agevole giocare se non si capisce discretamente la lingua. A questo proposito però constatiamo con piacere che la Leader distribuisce il programma dotato di manuale in italiano.

## Vindicators

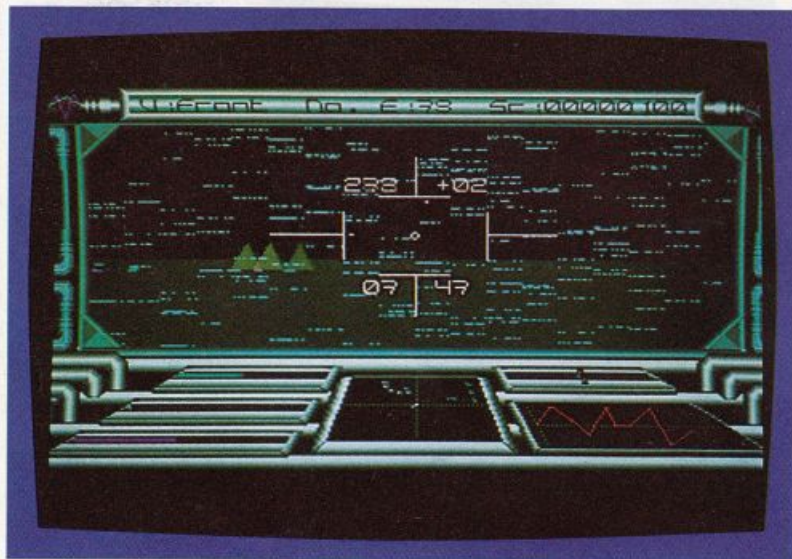
Il secondo gioco che vi presentiamo questo mese è decisamente più semplice e rilassante del precedente: si tratta della versione per Amiga di un noto Coin-op prodotto da una ditta poco conosciuta, la Tengen. Vindicators è il classico gioco spara e fuggi, tuttavia è sufficientemente originale e piacevole: iniziamo con le coordinate spazio-temporali. Vi trovate nell'anno 2525 su una stazione spaziale nei pressi della galassia TR15. La stazione sta subendo l'attacco delle forze dell'impero Tangent, e voi avete il compito di attaccare ed eliminare il nemico in modo da liberare tutti i livelli della stazione. Siete alla guida del carro armato strategico SR-88 e dovete giudarlo attraverso i vari livelli di gioco distruggendo tutti i carri armati nemici, nonché le torrette di protezione erette a difesa.

Il vostro carro è molto diverso dall'arma del nemico, infatti è molto più robusto e quindi resistente agli attacchi, tuttavia ha un'autonomia ridottissima, pertanto dovete fare spesso rifornimento di carburante. Lungo il percorso si trovano sia i fusti di benzina, sia un certo numero di stellette che vi servono per potenziare il carro. Il controllo del carro non è molto facile poiché, come un vero carro armato, non avete lo sterzo, bensì potete variare l'inclinzione del mezzo muovendo la barra a destra e a sinistra. Le vostre armi sono il cannoncino standard, più un certo numero di armi speciali selezionabili con il tasto Tab. Anche Vindicators è fornito di istruzioni in italiano, anche se non sono assolutamente necessarie. Vindicators è un gioco molto semplice, non è certo straordinario, tuttavia è piacevole ed è dotato di una grafica all'altezza di Amiga. Si tratta di uno dei tanti buoni giochi per Amiga senza infamia né lode.

*Millennium 2.2. La copertina del manuale d'istruzioni scritto in inglese, francese, italiano, spagnolo e tedesco.*

## Millennium 2.2

Questo titolo dal sapore di antica Roma nasconde un'avventura che al contrario si svolge nel 2200, esattamente nel mese di settembre. Vi potrebbe sembrare superflua la precisazione del mese, e infatti è proprio così, d'altra parte i tecnici della Electric Dream Software hanno redatto un'introduzione al gioco così dettagliata che abbiamo voluto tenere in considerazione la loro precisione. Non sono stati altrettanto precisi nello spiegare la dinamica del gioco, ma questo fa parte dell'avventura. Ma passiamo alla descrizione del gioco. Non vogliamo annoiarvi con l'incredibile storia che viene narrata per introdurre il gioco: vi basterà sapere che il pianeta Terra è stato distrutto dall'impatto di un enor-

me asteroide e pertanto l'ultimo baluardo di razza e cultura umana è costituito dalla popolazione di Base Luna 1.

Base Luna 1 è stata creata per scopi di ricerca, ma le mutate condizioni impongono un radicale cambiamento dell'uso della base. Da centro di ricerca, Base Luna 1 diventa il punto di partenza di una ondata coloniale che punta a trasformare altri pianeti in un ambiente terrestre. Naturalmente voi siete al comando di Base Luna 1 e quindi dovete prodigarvi innanzitutto per garantire la vita all'interno della base , ora che manca l'appoggio da Terra, e inoltre per organizzare la nuova ondata coloniale.

*Voyager. Vista dall'interno della navicella. L'immagine del paesaggio esterno è disturbata perché la nave si trova sotto il fuoco nemico.*

Il controllo della base avviene interamente tramite mouse: nella parte superiore dello schermo sono sempre visibili dieci icone che rappresentano tutte le operazioni che potete effettuare. La base è inoltre divisa in sette reparti, a cui potete accedere sempre servendovi del mouse. Complessivamente avete a vostra disposizione un completo servizio di informazioni e di comunicazione con le navicelle e con le colonie, nonché alcune opzioni che semplificano la simulazione. Tramite questi controlli dovete riuscire a effettuare la colonizzazione e mantenere attiva la base. La grafica del programma è molto semplice, ma buona. Millennium 2.2 comunque ci sembra il classico fumo senza arrosto, poiché dalla lettura delle istruzioni credevamo di avere a disposizione un ottimo programma, invece si tratta solo di un prodotto di poco interesse.

## Voyager

Terminiamo la nostra passerella tra le novità per Amiga con un prodotto di una software house di grande esperienza del campo dei videogiochi, la Ocean Software. Voyager è un altro gioco spaziale, ma molto più tradizionale di Millenium 2.2 poiché si tratta di un sofisticato spara e fuggi. Anche in questa occasione vi risparmieremo la narrazione della storia su cui si basa il videogioco. Voyager non è certo un gioco molto originale, però è realizzato molto bene e sarà senz'altro apprezzato dagli appassionati di giochi molto ricchi di armi, di controlli e di grafica tridimensionale.

Il gioco è molto semplice: dovete distruggere tutto quello che vi capita a tiro. Per fare questo però avete a disposizione un carro armato molto sofisticato e pensiamo che il divertimento maggiore sia costituito dall'imparare a sfruttarne tutte le caratteristiche. La grafica del programma non è eccelsa, ma è molto scenografica: nel complesso ci è piaciuta. Abbiamo apprezzato la possibilità di controllare il carro armato sia con il joystick, sia con il mouse. Lo scopo del gioco? Inutile dirlo: salvare la terra!

## Giochi 64

Questo mese lasceremo spazio a coloro che vedono nei videogame una fonte di divertimento immediata che non necessita di voluminosi manuali di istruzioni. Del resto sono proprio queste le persone che frequentano più assiduamente le sale giochi e che non disdegnano, nel caso se ne presenti l'occasione, una bella partita a bigliardo, a flipper o all'intramontabile calcetto.

Proprio a questi ultimi e ad altri tre giochi del genere si ispira American Club Sports, una raccolta dei passatempi più diffusi della Mindscape Inc.

I giochi sono caricabili indipendentemente l'uno dall'altro anche se un tabellone generale riassume i punteggi che i giocatori, quattro al massimo, hanno

conseguito nelle varie discipline. Vediamo quali sono gli aspetti salienti e le particolarità di ogni gioco:

• **Foosball (calcetto)**: è la riproduzione su computer del gioco più diffuso nei bar e nelle sale giochi in Italia. Il joystick permette tutti i movimenti delle aste con i giocatori e la loro selezione. Dal menù iniziale è possibile fare le proprie scelte sul numero dei giocatori (1 o 2), sulla velocità della pallina, sull'abilità del computer come avversario, sulla presenza o meno del cambio campo, sui goal da fare per vincere la partita, sul numero di partite per vincere il torneo.

• **Billiards (bigliardi)**: si usa il plurale poiché vi sono due tipi di bigliardo: quello inglese e il "Carom", ognuno dei quali si gioca con tre biglie ma con regole diverse. Si può scegliere il numero dei giocatori, quale concorrente deve iniziare, l'intensità dell'attrito, le condizioni di vittoria, il colore del tappeto, l'abilità del computer come avversario, il numero di partite per aggiudicarsi il match.

•**3-D Pinball (flipper)**: si tratta di un vero e proprio flipper in miniatura con bonus, extra balls e cancelletti apribili. Il giocatore può decidere l'angolazione del tavolo, il punteggio da raggiungere per vincere una pallina, il numero delle palline a disposizione, il numero di joystick da usare (1 o 2) e addirittura l'inserimento o meno del tilt che si attiva se si muove il joystick troppo vorticosamente.

• **Skeeball**: si tratta di un gioco diffuso specialmente nei lunapark e consiste nel centrare dei fori praticati in un tavolo con delle biglie. Ad ogni foro corrisponde un punteggio. Il giocatore può variare il grado di difficoltà o decidere se giocare contro il tempo, o avendo a disposizione un certo numero di palle.

• **Shooting Gallery (tiro a segno)**: è il tipico gioco dei parchi d'attrazione o delle fiere ambulanti; quattro schiere di animali scorrono sullo schermo di gioco e devono essere colpite dal giocatore; gli animali più veloci e più lontani procurano un maggior numero di punti. Il tiratore può scegliere il livello di abilità del computer se usato come antagonista, la velocità di movimento, il peso e il rinculo del fucile, il numero di tiri per partita o, se si preferisce, la durata della gara, l'attivazione o meno del sottofondo musicale.

• **Crazy Pool**: si tratta di un particolare bigliardo il cui obbiettivo consiste nell'imbucare sette biglie colorate servendosi di quella bianca. Le buche sono disposte al centro del tavolo e sono circondate da ostacoli che ne rendono difficile l'accesso. Il percorso per raggiungere le buche può essere variato a piacimento grazie al construction set inserito nel programma. Il giocatore può anche intervenire sul colore delle biglie, sul fattore d'attrito del tappeto e sul numero di partite da aggiudicarsi per vincere il match.

***Journey To The Centre Of The Earth. In viaggio verso l'ignoto.***

Le informazioni sopra elencate vi avranno già permesso di farvi un'idea sui giochi di American Club Sports. Per quanto riguarda un giudizio generale si deve sottolineare l'elevata giocabilità, dovuta non soltanto alla presenza di sei giochi in un solo programma ma anche alla possibilità di spaziare all'interno dei singoli passatempi variando le condizioni di partenza. La grafica è piuttosto elementare e anche il realismo dei

movimenti delle biglie o dei proiettili è un po'approssimativo.

### Naw Seal

Passiamo ad un gioco d'azione introducendo Navy Seal, creato dalla Cosmi Corporation. Siete il miglior incursore del gruppo speciale d'attacco della marina statunitense; duri mesi di addestramento vi hanno insegnato a fronteggiare qualsiasi situazione e pericolo, è arrivato il momento di fare sul serio: vi aspettano quattro pericolosissime missioni che devono assolutamente essere portate a termine per il bene del vostro paese. La prima consiste nel fare esplodere depositi di vitale importanza per il nemico, la seconda nel disinnescare le testate atomiche contenute in un sottomarino andato in avaria e affondato, la terza nel far saltare in aria una piattaforma petrolifera nemica, la quarta nel liberare un ambasciatore tenuto in ostaggio. Ogni missione si articola in diverse fasi, ognuna delle quali presenta numerose difficoltà: si parte dall'avvicinamento alla zona dell'operazione per via aerea o subacquea, si raggiunge l'obbiettivo in territorio ostile e, dopo aver eseguito l'ordine, bisogna salvare la pelle ritornando al campo base. In queste fasi avrete bisogno di tutto quello che avete imparato durante l'addestramento: precisione nel lancio col paracadute, destrezza nel muovervi sott'acqua, sangue freddo e riflessi pronti nel far fronte ad eventuali attacchi nemici servendovi delle armi in dotazione, velocità d'esecuzione e di fuga dalle zone pericolose. Seal si distingue dagli altri giochi del genere per la varietà degli obiettivi e per quella delle azioni che bisogna compiere per raggiungerli: in ogni fase ci si trova in una situazione diversa, e diverse sono le capacità sulle quali bisogna fare affidamento per uscirne sani e salvi. Inoltre la suspence dovuta al fatto che la missione possa sfumare in qualsiasi momento fa in modo che l'interesse aumenti con il procedere del gioco. A completamento del tutto vi è un test d'allenamento che il giocatore può compiere per saggiare le proprie capacità e vedere se è pronto per passare all'impiego operativo. Questo test è composto da una prova di tiro, da due percorsi ad ostacoli, uno su terraferma uno subacqueo) e da un'arrampicata.

***Navy Seal. Alcune videate dalla grafica molto attraente tratte dalle missioni proposte da questo gioco.***

La grafica è eccezionale ed è questa la lacuna più evidente del gioco. Una maggior cura in questo senso avrebbe potuto fare di Seal un ottimo programma nel genere d'azione.

### Journey to centre of the Earth

Il mondo dei videogames si fa sempre più colto; il mese scorso abbiamo parlato di un programma ispirato ai racconti di Tolkien e questa volta non ci smentiamo presentando Journey to the centre of the Earth, un adventure della Chip Softgold tratta dal celebre libro di Giulio Verne. Siete un illustre scienziato e avete l'obbiettivo di raggiungere il centro della terra per dimostrare la validità della teoria di un eminente collega. Il vostro è un viaggio verso l'ignoto che metterà a dura prova il vostro fisico non solo per la stanchezza e la carenza di cibo ma anche per gli imprevisti che accadranno quando meno ve lo aspetterete. Per questo sul quadro di gioco ci sono degli indicatori che vi permetteranno di

tenere continuamente sotto controllo la vostra condizione di salute. Il gioco finisce in caso di morte, per qualsiasi motivo essa sopraggiunga. Anche le riserve di acqua e cibo sono monitorate; i viveri sono sufficienti per un mese (questo vuol dire che bisogna raggiungere l'obbiettivo prima di tale periodo), mentre l'acqua può essere recuperata dalla condensa delle grotte in cui vi aggirerete. In caso di necessità c'è la possibilità di razionare i viveri o di compiere una sosta se la stanchezza vi sta debilitando il fisico in modo troppo elevato. Gli imprevisti, che possono capitare ad ogni istante, sono rappresentati da arcade, che appaiono improvvisamente e che richiedono prontezza di riflessi. Essi consistono nell'evitare massi che si distaccano dalle pareti delle caverne oppure orde di mammut che rischiano di investirvi.

Journey to the centre of the Earth è un buon programma e fa dell'originalità la sua qualità migliore. La grafica non fa sicuramente rimanere a bocca aperta, specialmente negli arcade, ma è apprezzabile l'idea di rappresentare la strumentazione in uno stile un po' antiquato, come richiede l'ambientazione storica della vicenda (ci troviamo infatti nel 1870). Anche se non è la prima volta che viene sfruttata , è buona la trovata di raffigurare le condizioni di salute del protagonista tramite la sua effige che, con l'aumentare dei disagi, appare sempre più stravolta e deperita. Interessante l'idea di poter scegliere il personaggio che ci rappresenterà sullo schermo di gioco fra quattro eminenti scienziati realmente esistiti; selezionandone uno appare una schermata con informazioni sulla sua vita e sui suoi meriti scientifici.

## Hostile all terrain Enounter

Semplicissimo come meccanica di gioco ma piacevole come grafica e giocabilità è H.A.T.E. (Hostile all-terrain enconter), un programma spaziale della Vortex Software. Al comando di un caccia interstellare sorvolate alcune basi orbitanti per raccogliere le Plasma Cells, cellule di energia indispensabili alla vita del vostro pianeta. Come ogni gioco spaziale che si rispetti non mancano le forze nemiche che ostacolano la vostra azione: proiettili vaganti, missili telecomandati, caccia nemici, barriere minate e ostacoli semoventi non vi renderanno la vita molto facile. Se il caccia viene colpito mentre sta portando una cellula, essa sarà distrutta al posto del veivolo. Più cellule si portano alla fine di un livello di gioco, più vite si avranno a disposizione in quello successivo. I livelli sono trenta: il superamento dei primi dieci vi classificherà "pilota di caccia", quello dei primi venti "comandante". H.A.T.E. fa parte di una lunga schiera di giochi spaziali non particolarmente pretenziosi ma ottimamente giocabili, in accordo con la premessa fatta all'inizio della recensione. Il gioco gode di una grafica tridimensionale che ne accresce l'effetto estetico. Molto utile per il superamento di punti ostici è la possibilità di utilizzo di una smart bomb, ordigno che distrugge ogni appostamento nemico nella zona del caccia. Nonostante questo aiuto non si è capito il motivo per cui i creatori del gioco abbiano dato al caccia un'unica vita: in questo modo il principiante è costretto a innumerevoli partite di pochi secondi prima di prenderci la mano e superare il primo livello; per lo più la schermata di preparazione al gioco che si ripropone ad ogni inizio è tutt' altro che breve.

**Gianni Arioli**
**Massimiliano Del Rio**

*American Club Sports. Il flipper. La dinamica di questo game lo rende addirittura migliore di quello reale!*

*Vi piacerebbe introdurre i vostri programmi con una schermata grafica in alta risoluzione multicolore? Oppure presentare le vostre videocassette con grafica e testo in scrolling? Se siete appassionati di computer grafica e di videoregistrazione non potete assolutamente non amare Message Maser e Picture Start*

# Videomania alla ribalta

La videomania ha raggiunto ormai tutti gli utenti del Commodore 64, e il livello grafico del software per questo famosissimo home computer è arrivato a livelli quasi professionali. Su questa rivista non potevano mancare utility volte alla creazione di show grafici. Questo mese pubblichiamo due ottime utility, adattissime per una eventuale videoregistrazione amatoriale. La prima consente di visualizzare messaggi in doppia dimensione e parzialmente in movimento, mentre la seconda offre la possibilità di visualizzare in modo originale una qualsiasi schermata in alta risoluzione personalizzata.

### Message Master

Con questa utility è possibile produrre messaggi elettronici piuttosto particolari. Lo schermo appare diviso in due par-

ti: la parte inferiore è fissa e può contenere un testo dai caratteri lampeggianti, a vari colori, e con dimensione doppia rispetto al normale.

Lo stesso tipo di caratteri può essere destinato alla parte superiore dello schermo, nella quale appare in movimento da destra a sinistra secondo uno scrolling fluido e ondulatorio. Lo sfondo dell'area di scrolling è costituita da fitte barre orizzontali colorate come in tutte le presentazioni di qualità per Commodore 64.

## Come si usa

Message Master è sulla cassetta allegata alla rivista. Per caricarlo e lanciarlo basta che digitiate :

LOAD"MESSAGE MASTER"

dopo aver posizionato il nastro opportunamente. Premete Run al termine del caricamento. Siccome il programma funziona regolarmente solo su disco, è utilissimo servirsi dell'utility Dsave V2 (vedi riquadro sul trasferimento dei programmi da cassetta a disco). Al termine del caricamento, comunque, prima di effettuare il Run, potete salvare il programma su disco anche con un normale Save.

Dopo aver effettuato il lancio del programma date volume e godetevi la gradevole presentazione. La pressione della barra spaziatrice vi conduce alla prima sezione del programma.

## Testo Fisso

La presentazione finale è costituita da un testo fisso nella parte bassa dello schermo e da una parte mobile nella parte soprastante. A questo punto del programma dovete decidere il contenuto del testo fisso. I caratteri della presentazione finale hanno una dimensione doppia rispetto al normale. Per questa ragione, in fase di editazione, ad ogni carattere che digitate corrisponde un gruppo di quattro caratteri normali di vario tipo, fra i quali c'è anche il carattere corrispondente a quello digitato (vedi **figura** 1). Lo spazio che avete a disposizione in

*Figura 1. Message Master. Ecco la prima sezione del programma dove è possibile digitare la parte fissa del testo*

questa fase corrisponde esattamente alla zona riservata al testo fisso nello show finale. Le funzioni di editing abilitate in fase di digitazione corrispondono a quelle standard dell'editor Basic. Per passare alla fase successiva del programma occorre occupare completamente lo spazio a disposizione. Se il testo non arriva esattamente fino all'ultimo blocco disponibile (spigolo inferiore destro del quadro), potete completare l'editazione premendo ripetutamente la barra spaziatrice. Quando raggiungete l'ultimo blocco, il programma passa alla fase successiva dopo una breve pausa di working.

## Testo in scrolling

Come abbiamo già accennato, una parte della presentazione finale è costituita da testo in movimento. Questo movimento si articola in uno scrolling orizzontale da destra a sinistra e in un moto armonico verticale. In pratica, la combinazione di questi due movimenti si traduce in un percorso sinusoidale molto gradevole globalmente, ma un po' scomodo per l'occhio di chi vuole leggere il testo. D'altro canto, siccome nessuno vi obbliga a porre esclusivamente del testo in questo spazio movimentato, e siccome il set di caratteri in doppia dimensione include un gran numero di simboli grafici, è sicuramente possibile inventare qualche struttura grafica adatta al movimento.

Il testo di scrolling può essere lungo diverse migliaia di caratteri. Le funzioni di editing abilitate in questa fase del programma sono quelle di cui si dispone

quando ci si trova fra due doppi apici, usando l'editor standard del C64 (vedi **figura 2**).

Per questa ragione i tasti cursore sono disabilitati e alla loro pressione corrisponde un simbolo grafico. I tasti funzione, invece, offrono una prestazione particolare. Infatti, il simbolo grafico che compare alla pressione di uno di questi tasti determina la velocità orizzontale dello scrolling. Il simbolo grafico di funzione può essere inframmezzato al testo come un carattere qualunque. Il tasto F1 imposta la velocità minima, mentre il tasto F5 la massima. Osservate che i simboli grafici dei tasti funzione non vengono visualizzati nel testo dello show finale ma, semplicemente, variano la velocità di scrolling alla comparsa dei caratteri adiacenti ad essi.

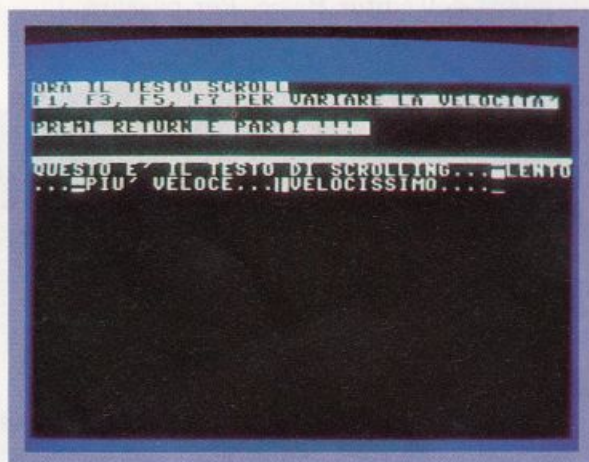



*Figura 2. Message Master. Questa è la seconda fase del programma. Il testo di scrolling può essere lungo migliaia di caratteri*

La pressione del tasto Return decreta la fine del testo di scrolling, che sarà mostrato a ripetizione nella presentazione completa. Per questa ragione consigliamo di porre dei caratteri spazio al termine del testo di scrolling prima di premere il tasto Return, in quanto il programma non pone alcuno spazio tra l'ultimo carattere del testo e il primo.

## Salvataggio presentazione

Premendo Return vi trovate nella fase successiva. A questo punto non vi resta che procurarvi un disco formattato, e con almeno un centinaio di blocchi liberi, inserirlo nel drive e premere la barra spazio. Quando le operazioni di salvataggio sono concluse potete resettare il computer e caricare la directory del disco. Muovete il cursore sul file che si chiama Paster. Digitate LOAD sovrapponendovi al numeo di blocchi, spostate il cursore verso destra, appena oltre le virgolette di chiusura, digitate il solito ",8,1" e premete Return. Al termine del caricamento azzerate lo schermo con Shift Clr/Home e digitate SYS 5888. Infine premete Return.

Se tutto è in regola dovreste vedere la vostra presentazione come in **figura 3**. Il testo fisso viene visualizzato in un modo particolare in un secondo tempo.

## Compattamento

Sul numero di maggio 1989 di *Radio Elettronica & Computer* abbiamo pubblicato un'utility che può rivelarsi assai utile per chi volesse avere la presentazione compattata in un file unico lanciabile con un normale Run. La prima cosa da fare è digitare la seguente linea Basic, dopo aver resettato il computer:

```
10 SYS5888
```

Premete Return per confermare in memoria la linea di programma e poi salvatela su disco con SAVE"LINEA",8.

L'utility, che si chiama Gigapack, va lanciata per prima, dopo aver resettato ancora il computer. In seguito si deve dare un New e caricare il file di presentazione come illustrato sopra, ma senza assolutamente richiamare la directory del disco. A questo punto, senza dare alcun altro comando, dovete caricare la linea Basic che avete digitato e salvato in precedenza.

Ora, come è stato ampiamente spiegato sul numero di maggio, attivate Gigapack con SYS820 e poi salvate il tutto su disco o nastro. In questo modo le vostre presentazioni possono anche essere salvate su cassetta senza problemi.

## Videoregistrare

Per chi volesse videoregistrare la presentazione, consigliamo di porre un cer-

to spazio all'inizio del testo di scrolling, per poter sincronizzare più facilmente l'inizio della registrazione con l'inizio dello show.

Per quanto riguarda l'interfacciamento fra computer e videoregistratore, sconsigliamo di effettuare il collegamento tramite l'uscita TV del computer, ma suggeriamo di rivolgersi a un rivenditore di apparecchiature Commodore per effettuare al meglio i collegamenti mediante connessioni appropriate.

## Picture Start

La seconda utility che trovate su questo numero, anch'essa volta alla creazione di presentazioni e introduzioni, consente di porre una schermata in alta risoluzione (formato Koala) prima di un programma. In pratica genera un file unico che include sia la schermata sia il programma in Basic, o in linguaggio macchina.

La schermata appare subito dopo il lancio del file definitivo, ed entra in scena in un modo alquanto singolare: suddivisa in strisce orizzontali, alte otto pixel ciascuna, si costruisce per aggregazione di queste sezioni di grafica che, scrollando dal basso verso l'alto, formano l'immagine completa. Dopo qualche istante la schermata si decompone secondo lo stesso meccanismo inverso, per ricomporsi immediatamente dopo. L'intero processo avviene periodicamente ogni due o tre secondi.

Come sottofondo è stato definito un effetto sonoro che, inizialmente, è piuttosto sgradevole, ma diventa sempre più interessante con il passare dei secondi. La pressione della barra spaziatrice determina il lancio del programma principale.

## Come si usa

Anche questo programma può essere utilizzato solo da disco. La prima cosa da fare, quindi, è effettuare il trasferimento dello stesso, dalla cassetta al floppy, servendosi dell'utility Dsave V2, anch'essa sulla cassetta. Il passo successivo consiste nel procurarsi una schermata in formato Koala e un programma a cui agganciarla (se non sapete come generare tale schermata vi rimandiamo al numero 24 di *TuttoCommodore Fumetti* oppure al numero 33 di *Commodisk*, dove sono contenuti rispettivamente un programma per generare file grafici Koala e un programma per convertire altri formati in quello Koala). Fatto questo procedete al caricamento e al lancio del programma.

La prima schermata può essere superata premendo il tasto Ctrl. La schermata successiva visualizza un messaggio d'Input. Il nome che viene chiesto è quello del file grafico. Ricordate che i file Koala hanno un nome particolare. Per questa ragione il primo carattere che digitate deve essere un punto di domanda, mentre l'ultimo un asterisco. Quando il caricamento è avvenuto regolarmente (la

*Figura 3. Message Master. La schermata di presentazione in azione. Catturato al volo il testo di scrolling*

luce del drive non lampeggia ed è spenta), viene chiesto se si vuole vedere la schermata. Il tasto Y corrisponde a Sì. Per uscire dalla visualizzazione basta premere la barra spazio. Eseguita l'operazione viene chiesto se si desidera caricare un'altra schermata (premete il tasto Y per rispondere sì).

Una risposta negativa porta alla richiesta del nome del programma a cui volete allacciare la schermata. A questo punto Picture Start carica il programma e vi allaccia la presentazione. Al termine resta in attesa della pressione della barra spaziatrice. In questo momento lo schermo è tutto di uno stesso colore, il drive è in condizione di OK (led rosso spento) ed è possibile estrarre il disco per inserire quello

di destinazione. Premendo la barra spazio il file definitivo viene scaricato.

**Suggerimenti**

La struttura del programma consente di creare una sorta di Slide Show, ovvero una lunga presentazione costituita da varie schermate mostrate una dopo l'altra. Per ottenere questo effetto basta creare un file definitivo come spiegato sopra, e attaccarvi una schermata come fosse un programma qualsiasi. Il file definitivo ottenuto in questo modo può subire lo stesso trattamento generando un file definitivo costituito da tre schermate. Come si può facilmente immaginare, è possibile applicare questo procedimento indefinitamente, e l'unica limitazione è costituita dalla memoria del C64.

Per avere il maggior numero di schermate possibile ci si può servire ancora del programma Gigapack, menzionato nel paragrafo relativo al programma Message Master. Occorre compattare tutti i file definitivi man mano che vengono generati oppure, per rendere più agili le operazioni, compattare un file definitivo ogni tre o quattro file.

Al termine dello Slide Show si può porre un breve programma Basic, che carica e lancia di nuovo la lunga presentazione e consente uno spettacolo continuo.

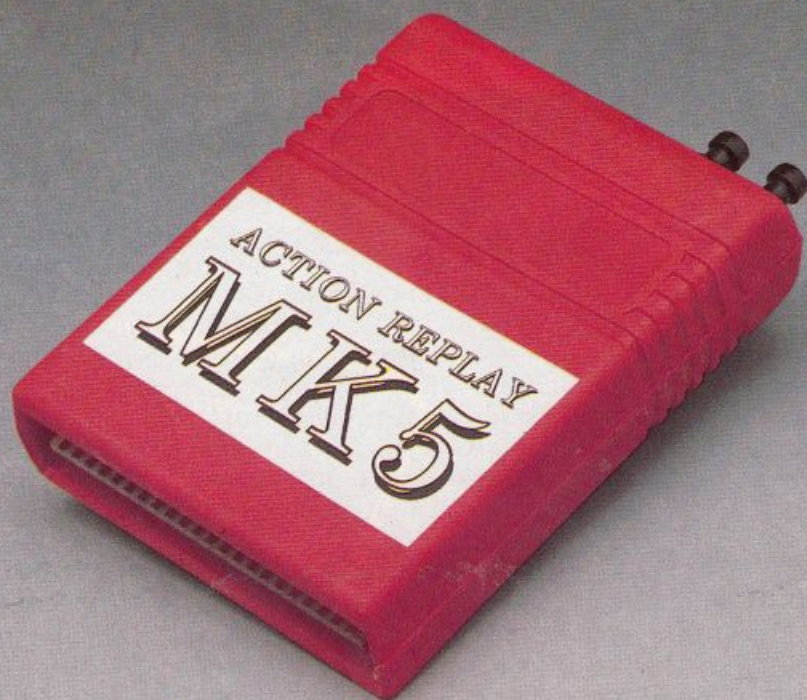
**Raffaele Zanini**

# L'ultima frontiera

*Già in distribuzione in tutta Italia, la nuova portentosa cartuccia Action Replay MK5 riscuote ovunque un brillante successo. Eccovi le nostre impressioni dopo una accurata analisi e un collaudo intensivo.*

Già da molto tempo non vi parliamo di cartucce per il C64. In realtà eravamo convinti che ormai era già stato fatto tutto il possibile per espandere il glorioso Commodore in tutti i modi, in particolare, alcuni mesi fa abbiamo provato per voi la Final Cardridge III e The Expert, e abbiamo affermato che si trattava di due cartucce davvero eccezionali. Non è nostra intenzione smentire ora tale affermazione, tuttavia ancora una volta dobbiamo prendere atto che esiste sempre la possibilità di migliorare: ecco a voi MK5 Professional, la più potente cartuccia in commercio.

MK5 è una cartuccia tuttofare per dilettanti e professionisti ed è finalizzata alla copia e alla modifica dei programmi, con un occhio particolare per i videogiochi.

Non pretendiamo però di anticiparvi in poche righe le numerose operazioni che può eseguire MK5, pertanto procediamo

con una descrizione dettagliata.

Come la quasi totalità delle cartucce per C64, anche MK5 si presenta come una piccola scatola di plastica dotata di un connettore per il bus del computer e di due piccoli pulsanti, uno di Reset e l'altro di Freeze.

All'accensione del computer, oppure dopo la pressione del tasto di Reset compare il primo menù, che raccoglie quattro opzioni di lavoro. La prima è un vero e proprio Reset che nasconde totalmente la MK5 al computer: questo significa che nessun programma può accorgersi della presenza della cartuccia dopo avere effettuato questa operazione. Il secondo Reset è apparentemente equivalente al primo, ma in realtà configura la memoria in modo tale che la MK5 è grado di compattare nel modo migliore un programma per effettuare il backup.

Prima di effettuare un backup, è necessario utilizzare uno di questi due Reset; normalmente è meglio usare il secondo, in modo da ottenere un migliore compattamento, tuttavia può accadere che alcuni programmi si rifiutino di girare correttamente, quindi si deve procedere con il primo.

La terza opzione del menù iniziale attiva il menù Utilities, di cui parleremo in seguito, mentre l'ultima opzione attiva la cartuccia in modo Fastload. Naturalmente non si tratta di un semplice fastload, bensì di un sistema di espansione davvero complesso, ma anche di questo parleremo in seguito.

Iniziamo invece a descrivere le operazioni di Backup e Modifica, che probabilmente interessano la maggior parte dei possibili utenti della MK5.

MK5 è in grado di congelare qualsiasi programma in qualunque momento, basta effettuare una delle due operazioni di reset, caricare il programma e premere il tasto Freeze. In questo modo compare il menù Freeze che è ricco di opzioni. Se volete semplicemente salvare il vostro programma vi basterà scegliere l'opzione Backup: in questo modo il computer provvederà a compattare il programma dopodiché vi mostrerà il menù Backup. Da questo menù è possibile scegliere tra i numerosi modi di salvataggio. Sono disponibili tre modi per il floppy disk e due modi per la cassetta.

Per il floppy disk è disponibile il salvataggio standard che utilizza le routine del kernal, ovvero il sistema operativo del C64. In questo caso i programmi di lunghezza superiore ai 202 blocchi vengono salvati in due file.

Questo salvataggio è sicuramente il più adatto a chi dispone di un velocizzatore parallelo tipo Speeddos, mentre non è assolutamente pratico negli altri casi. Il salvataggio turbo accelera notevolmente le prestazioni del drive: realizza un file singolo che viene caricato a una velocità davvero buona.

Ma non è finita qui: la terza velocità di save è davvero eccezionale. Si tratta del sistema Warp-25 che realizza un unico file di formato non standard che può essere caricato in soli 6-7 secondi. Sì, avete capito bene: un intero programma di 200-220 blocchi può essere caricato in una manciata di secondi.

Inutile sottolineare che si tratta di una prestazione davvero eccezionale, che però comporta alcuni svantaggi: infatti, poiché la Warp-25 non utilizza il formato standard del DOS, non è possibile utilizzarla con il 1571 in modo doppia faccia, né con il 1581, inoltre i file occupano una porzione maggiore del disco. È anche evidente che per utilizzare la Warp-25 è necessario avere un drive in ottime condizioni, ma comunque noi abbiamo fatto alcune prove e non abbiamo avuto mai alcun problema.

Per caricare un programma salvato in turbo, oppure in modo Warp-25, è indispensabile utilizzare il fast-load della MK5 oppure il loader, un breve programma che può essere salvato su tutti i dischi semplicemente selezionando un'apposita opzione.

Poiché stiamo parlando di dischi, è senz'altro il caso di spendere due parole per evidenziare un'eccellente possibilità offerta da MK5 a chi possiede il drive 1571. Com'è noto tale drive è in grado di leggere e scrivere su entrambe le facce del disco contemporaneamente, ma solo se viene gestito da un C128, in modo 128. La MK5 però prevede un comando che consente al 1571 di leggere entrambe le facce contemporaneamente anche utilizzando un C64 oppure un C128 in modo 64. Si tratta di una comodità davvero pregevole che consente di risparmiare tempo e spazio sui dischi.

Passiamo alla registrazione su nastro. Sono disponibili due modi: il turbo e il superturbo (manca invece un po' di fantasia nell'attribuire i nomi). Il primo modo salva a velocità 5-6 volte superiore allo standard, mentre il secondo a velocità 8-10 volte superiore. Anche in questo caso è chiaro che la velocità superiore richiede che il registratore sia in ottime condizioni, ovvero testina pulitissima e ben allineata e nastro di buona qualità.

Solitamente diamo un giudizio complessivo sulla cartuccia al termine della prova, ma in questo caso dobbiamo interrompere la descrizione della cartuccia, per affermare che i sistemi di registrazione e caricamento della MK5 sono davvero eccellenti.

Vediamo ora le altre possibilità offerte dal menù di Backup. Esistono diverse possibilità di gestione della grafica; ad esempio

c'è un ottimo sprite monitor, che consente di visualizzare tutti gli sprite contenuti in memoria al momento della pressione del tasto Freeze. Questo monitor consente anche la cancellazione di uno sprite, oppure il salvataggio e il caricamento di uno sprite. In questo modo se disponete di uno sprite editor potete modificare a vostro piacimento gli sprite di qualsiasi gioco.

Non poteva mancare il classico sprite killer, che disabilita le collisioni degli sprite. Come tutti gli sprite killer anche questo prevede tre possibilità, ossia disabilitare le collisioni tra sprite e sprite, tra sprite e fondo, oppure entrambe le opzioni contemporaneamente. Ciò consente a molti giochi la possibilità di non morire mai.

Anche la grafica in alta risoluzione è stata considerata, poiché esiste un comando che salva le schermate hi-res multicolor (generalmente le schermate di presentazione dei giochi) nei formati dei programmi grafici più noti. È anche possibile stampare direttamente le schermate grafiche: basta disporre di una stampante Commodore, o anche di una stampante grafica Epson compatibile con porta parallela. In quest'ultimo caso, è indispensabile utilizzare un apposito cavo per collegare la porta del C64 alla stampante.

A questo proposito possiamo aggiungere che, usando tale cavo, MK5 consente l'uso delle stampanti Epson compatibili con il C64 anche per stampa di caratteri, infatti è in grado di effettuare la necessaria conversione dei codici. Si tratta di un'opzione molto utile a chi possiede anche un'Amiga o un IBM compatibile, poiché consente di utilizzare un'unica stampante per entrambi i sistemi.

Abbiamo parlato di sprite e di alta risoluzione, ci manca solo il modo testo. MK5 dispone anche di un test editor per modificare le schermate di testo. Si tratta di un vero e proprio editor di schermo, che consente l'elaborazione della schermata dove il programma è stato congelato.

Terminata la parte grafica vediamo una delle caratteristiche più eclatanti di MK5: l'inseritore di poke e soprattutto il Pokefinder, ovvero il cercatore di poke. Già da molto tempo capita di vedere su alcune riviste specializzate degli elenchi di poke- che servono per dare al programma le vite infinite, ossia conferire l'immortalità al protagonista del gioco- oppure delle poke, che permettono di iniziare dallo schema desiderato e così via. MK5 consente il facile inserimento di tali poke nel programma, inoltre, poiché è possibile salvare il programma dopo avere effettuato tale operazione, in sostanza MK5 vi permette la facile realizzazione di giochi di tipo trainer.

mente, siano esse codici ASCII o codici di schermo, e non manca un completo disk editor che consente la modifica dei singoli blocchi dei floppy disk.

Ci resta da vedere cosa succede quando dal menù iniziale si seleziona l'opzione Fastload. Così facendo si effettua un reset che lascia impostato un sistema operativo molto più ricco dello standard. I tasti funzione vengono attivati, ed è possibile utilizzare i diversi sistemi di caricamento programmi inoltre anche il Basic risulta espanso con utili comandi. Non dimentichiamo la possibilità di gestire il drive 1571 in modo a doppia faccia (ben 1328 blocchi su un disco) e tanti altri comandi.

Ma non è finita qui, anzi, il meglio deve ancora venire, poiché MK5 è anche in grado di trovare da sola le poke per realizzare le vite infinite.

Si tratta davvero di una possibilità degna di nota; se avete un gioco che vi appassiona e non riuscite mai a superare un certo schema perché venite sistematicamente uccisi non dovete far altro che usare pokefinder e nessuno vi potrà più uccidere.

Naturalmente questo procedimento non funziona sempre. Nelle istruzioni della MK5 si parla di successo nel 90 per cento dei casi, ma a noi dopo avere effettuato qualche prova tale cifra pare eccessiva, poiché molti programmatori usano sistemi non standard per gestire il numero delle vite. Questo non toglie che Pokefinder sia uno strumento davvero utile e potente.

Un'altra opzione offerta dal menù Backup è il monitor. Si tratta di un monitor davvero molto completo che consente ogni tipo di manipolazione. Non ci attarderemo nella descrizione di tutti i comandi, vi basterà sapere che; tra le altre cose, è possibile impartire comandi Basic, è possibile modificare parti di testo contenute in memoria direttamente, siano esse codici ASCII o codici di schermo...

Siamo davvero entusiasti di questa cartuccia: MK5 è il non plus ultra sul mercato, e soprattutto è molto efficiente e precisa. Non possiamo che consigliarne l'acquisto a chiunque ami mettere le mani all'interno dei programmi, poiché è il massimo dei toolkit.

È possibile acquistare MK5 presso Flopperia srl, a Milano in via Montenero, 31. Tel. 02/55180484 oppure in tutti i migliori rivenditori hardware e software della vostra città.

### Flash Fire a sensori

Questo mese vi vogliamo parlare di un accessorio che spesso passa in secondo piano, forse perché è molto semplice e poco costoso, quindi capita di non badarci eccessivamente: il joystick.

L'occasione ci è offerta da Flash Fire a sensori. La prima cosa che ci ha colpito è sicuramente la confezione: si tratta di una scatola molto grande rispetto alle dimensioni dell'accessorio, molto variopinta. Il contenuto è ancora più strano, poiché oltre a Flash Fire a sensori si trova una simpatica busta di nylon adatta per il trasporto del prezioso strumento. Certamente una confezione molto apprezzabile.

In pratica si tratta di un joystick in cui la consueta leva di controllo è sostituita da quattro sensori. Ovviamente anche il solito tasto Fire è stato realizzato mediante un sensore.

Come si può vedere dalla foto, Flash Fire a sensori non somiglia assolutamente a un joystick convenzionale, infatti non è altro che una piccola scatola rettangolare piatta dotata di sei sensori, che sono delle piastrine di plastica trattata apparentemente insignificanti. Quattro piastrine sono disposte a quadrato e servono per il controllo della direzione, mentre le altre due sono i tasti di Fire.

Flash Fire a sensori è dotato della solita presa joystick, compatibile con la maggioranza dei computer, ma poiché si tratta di un sistema un po' insolito i progettisti hanno avuto l'ottima idea di dotarlo di una presa passante per un joystick convenzionale. Questo significa che è possibile utilizzare Flash Fire a sensori, oppure un altro joystick, senza dovere mai staccare le spine di connessione.

Poiché il computer vede il joystick come un insieme di pulsanti, Flash Fire a sensori è dotato nel proprio interno di un'elettronica in grado di tradurre gli impulsi elettrici, forniti dal contatto dei sensori, in segnali adatti a una presa joystick. L'elettricità necessaria per il circuito viene

prelevata direttamente dal computer. Si tratta di un circuito molto semplice, composto da due circuiti integrati e poca componentistica. I progettisti hanno un evidente timore di vedere copiato il loro lavoro, infatti le sigle dei circuiti integrati sono state cancellate con la carta vetrata.

Terminata la descrizione generale possiamo passare alle sensazioni d'uso. Abbiamo provato Flash Fire a sensori con diversi videogiochi e la prima impressione è una grande differenza rispetto ai sistemi usuali. È necessario un minimo di pratica per potere sfruttare efficacemente Flash Fire a sensori.

Dopo il primo rodaggio siamo in grado di dire che questo rivoluzionario sistema di controllo presenta alcuni vantaggi nonché alcuni difetti.

Tra i vantaggi dobbiamo annotare la superiore velocità e precisione; l'assenza di organi meccanici, cancella ogni tipo di ritardo dovuto allo spostamento dei contatti. Anche la sensazione tattile è molto piacevole, in quanto ci si accorge di avere un controllo molto sensibile sullo strumento. Non dimentichiamo inoltre che l'assenza di leve e contatti rende Flash Fire a sensori indistruttibile, a meno di trattarlo in modo assolutamente inusuale.

Dobbiamo anche registrare un difetto, che probabilmente avete già intuito. Il controllo di Flash Fire a sensori è senza dubbio meno intuitivo del controllo di un joystick convenzionale, e non può essere diversamente, poiché qui non si tratta di muovere una leva nelle otto possibili direzioni, ma bisogna imparare a toccare i sensori giusti per ottenere l'effetto desiderato.

Questo non significa che Flash Fire a sensori fornisca prestazioni inferiori agli altri joystick, bensì significa che è necessario utilizzarlo a lungo prima di poterlo gestire al meglio.

**Gianni Arioli**

*Il joystick Flash Fire a sensori è disponibile presso GP Elettronica, via 4 novembre 32/34, 20092 Cinisello Balsamo (MI), tel. 02-6189551.*

## *Per trasferire i programmi di* RE&C *da cassetta a disco*

*Su ogni numero di* Radio Elettronica & Computer *trovate l'utility Dsave V2, registrata subito dopo il file d'intestazione. Questo programma consente di trasferire la maggior parte dei programmi della cassetta su un floppy disk. Gli unici programmi che non è possibile trasferire sono quelli che, per ragioni di copyright, sono salvati sulla cassetta in modo autostart.*

*Il programma Dsave V2 è d'uso semplicissimo: dopo il caricamento del solito menù all'inizio della cassetta si deve caricare e lanciare Dsave v2 con LOAD"DSAVE*". Il menù che compare sul video offre tre possibilità:*

*1 - La cassetta verrà letta e il primo programma incontrato caricato. A questo punto viene chiesta conferma per il trasferimento sul disco, dopodiché si passerà al caricamento del successivo programma sulla cassetta e cosi via.*

*2 - Scegliendo la seconda opzione, invece, verranno salvati su disco tutti i programmi automaticamente, senza selezioni da parte dell'utente. A questo proposito bisogna ricordare che, a volte, sulla cassetta vi sono dei file sequenziali o programmi particolari che il copiatore non può trasferire, nel qual caso il processo si blocca. Vi consigliamo quindi di utilizzare l'opzione 1 anche se volete trasferire tutti i programmi della cassetta.*

*3 - Questa opzione consente di visionare la directory del disco.*

# Ricorsività, grafica, calcoli e numeri

*Entriamo, con questa puntata, nel cuore dell'argomento Logo. Ricorsività avanzata e conseguenti applicazioni grafiche, sono le prime battute di una approfondita trattazione sui più affascinanti problemi della programmazione.*

CAPITOLO UNDICESIMO

## Quale Storia?

*C'era una volta un re, seduto sul sofà, che disse alla sua serva: "Raccontami una storia!" e la serva cominciò: "C'era una volta un re, seduto sul sofà, che disse alla sua serva: "Raccontami una storia!" e la serva cominciò:"C'era una volta...*

Tutti i bambini sanno come finisce questa storia, o meglio come non finisce. Non è difficile vedere che nessun narratore mortale può riuscire a raccontarla tutta, indipendentemente dalla pazienza dei suoi ascoltatori. Eppure, qualunque narratore mortale saprebbe esattamente come proseguirla a partire da qualsiasi punto, dopo averne ascoltato solo un breve stralcio come quello qui sopra. Egli, come voi, avrà semplicemente notato che per andare avanti è sufficiente applicare una certa regola, chiamiamola regola ricorsiva, che potremmo riassumere così: la storia viene raccontata all'interno di se stessa.

Se pensiamo ad una storia come a qualcosa che sta racchiuso tra virgolette, sappiamo anche che essa finirà nel momento in cui chiudiamo le virgolette che avevamo aperto all'inizio. Ecco perché non riusciamo ad arrivare in fondo: prima di avere il tempo di chiudere un qualsiasi paio di virgolette, ecco che la regola ricorsiva ci costringe invece ad aprirne ancora. Provate a chiedere ai vostri amici quanti re ci sono nella storia. Alcuni vi diranno che ce ne sono un'infinità; un amico arguto e sospettoso, sentendo odore di domanda-tranello, potrebbe, però, rispondervi così: "C'è soltanto un re, il primo. Gli altri sono solo raccontati."

La cosa divertente è che tutti i re della storia, se interrogati, potrebbero darvi la stessa risposta, riferendosi al re immediatamente successivo: il re di livello uno (il primo) sta infatti ascoltando una storia che parla del re di livello due, e il re di livello 4 ne ascolta una sul re di livello 55.

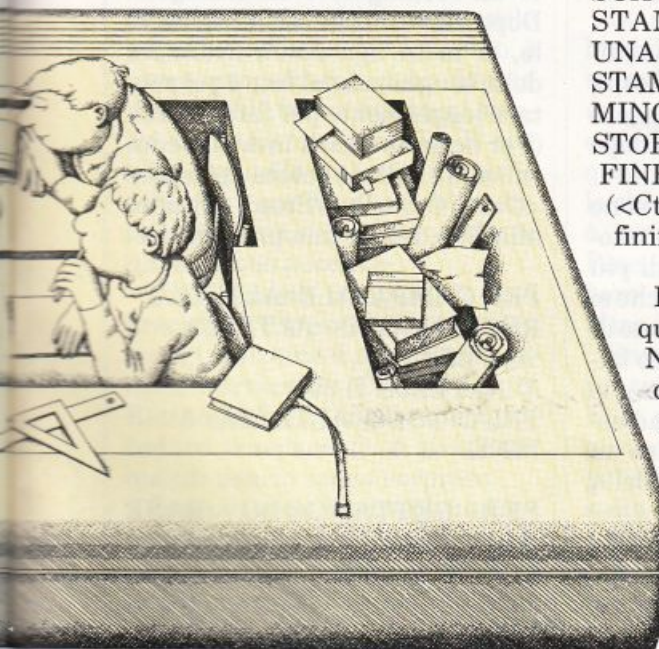
Se ci pensate un attimo, vi renderete conto che le storie sono tut-

te uguali. Ma questo non lo avevamo già detto?

Andate ora dal vostro amico più furbo, che avevate lasciato appositamente per ultimo. Iniziate la tiritera di "c'era una volta un re" eccetera, e quando vi interrompe domandategli a bruciapelo: "Quanti re ci sono in questa storia? "Quale storia?" vi risponderà e adesso nei guai siete voi: non potete certo raccontargliela tutta!

Per mostrargli di quale storia state parlando, e fargli così venire un discreto mal di testa, non avete altra scelta che sfoderare la regola ricorsiva di poche righe fa: la storia viene raccontata all'interno di se stessa. Ma qui non si parla affatto di re! E allora?

Potete sempre applicare quella specie di regola generale a questa particolare storia, inserendo una chiamata ricorsiva all'interno della spiegazione. Vale a dire: date un nome alla storia e poi chiamate quel nome dall'interno della la storia stessa, in modo da farla ricorrere.

Prendete quindi un pezzo di carta e scrivete qualcosa di questo tipo:

La storia di cui parlo si chiama Storia ed è fatta così:
aperte le virgolette
C'era una volta un re, seduto sul sofà che disse alla serva:
"Raccontami una storia!"
e la serva cominciò:
E qui la serva gli raccontò la Storia.
Chiuse le virgolette
Fine

Passate il foglio al vostro amico e uscitevene soddisfatti, mentre lui tenta disperatamente di chiudere le virgolette.

## Che cosa c'entra?

Alcuni mesi dopo il vostro amico, di cui nel frattempo avevate perso le tracce, vi invita a casa sua per mostrarvi il suo ultimo programma in linguaggio Logo. Si tratta di una singola procedura, che suona così:

```
PER STORIA
STAMPA[]
STAMPA ["C'ERA UNA VOLTA
UN RE]
STAMPA[SEDUTO SUL SOFÀ]
STAMPA [CHE DISSE ALLA
SUA SERVA:]
STAMPA ["RACCONTAMI
UNA STORIA!"]
STAMPA [E LA SERVA CO-
MINCIO:]
STORIA
FINE
```

(<Ctrl> C o <Run/Stop> per definire la procedura)

Premete Storia, e osservate quello che succede. "Vedi? Non riesco a fermarlo!", vi dice l'amico. Vi sembra di ricordare che ci sono tre modi di arrestare l'esecuzione di una procedura, e glieli spiegate:

- <Ctrl>- G: blocca tutto definitivamente con un clamoroso Stop!
- <Ctrl> -W: sospende l'esecuzione fino a quando non premete un tasto
- <Ctrl>- Z: provoca una Pausa, permettendovi di premere altri comandi e di riprendere poi l'esecuzione dal punto in cui l'avevate arrestata.

Per ripartire dovete premere Continua o la sua abbreviazione CO.

Lo stesso effetto di <Ctrl> -Z: si può ottenere mettendo all'interno di una procedura un'istruzione di Pausa.0:ln1

Li provate tutti e tre, e quello che attira la vostra curiosità sono i messaggi provocati da:

<CTRL>-G
<CTRL>-Z:PAUSA (oppure STOP!),
NELLA RIGA
STAMPA[SEDUTO SUL SOFÀ]
(per esempio)
DI STORIA AL LIVELLO 1273
(per esempio)

Evidentemente (c'è scritto), la storia di cui avete interrotto l'esecuzione era quella di 1273-esimo livello, a proposito del 1273esimo re, dopo che erano state aperte 1273 virgolette.

Se il primo re si trova al livello uno, chi sarà mai il re di livello zero?

Ma siete voi, che raccontate la storia, o meglio il vostro amico che la sta ascoltando. Provate premendo Pausa quando non c'è nessuna procedura in azione.

## Ferma!

Concorderete con noi sul fatto che le procedure di poco fa sono in un certo senso pessimi esempi di programmazione; non sembra molto utile che un programma continui per sempre ad eseguire la stessa sequenza di operazioni.

Ma se cambiassimo qualche cosa (un "parametro") ad ogni pas-

so delle nostre procedure, potremmo forse trovare il modo di fermarci.

Così, ricordando l'abitudine di certi sovrani di dormire su guanciali di seta rarissima, potremmo inventare:

C'era una volta un re con cinque guanciali che disse alla sua serva:
"Raccontami la storia di un re con un guanciale in più di me!"
E la serva cominciò:
"C'era una volta un re con sei guanciali..."

oppure:

```
PER SPIRALE :LATO
A :LATO
D 90
SPIRALE :LATO + 5
FINE
```

Nel primo caso il parametro è il numero dei guanciali (l'unica cosa che cambia tra i vari livelli della storia), nel secondo caso è :Lato, la lunghezza del lato. Osservate Spirale 10 con la Traccia attiva.

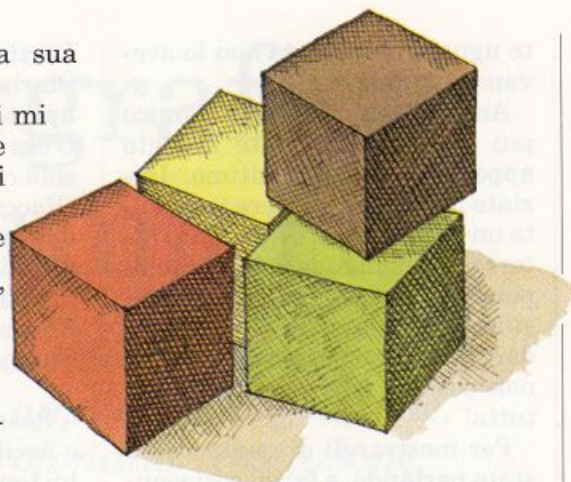
E per fermarsi? Basta che ad ogni giro controlliamo il verificarsi di una certa condizione...

C'era una volta un re con cinque guanciali che disse alla sua serva:
"Se avessi sette guanciali mi addormenterei ma visto che non li ho
raccontami la storia di un re con un guanciale in più di me!"
e la serva cominciò:
"C'era una volta un re con sei guanciali che disse alla sua serva:
"Se avessi sette guanciali mi addormenterei ma visto che non li ho
raccontami la storia di un re con un guanciale in più di me!"
E la serva cominciò:
"C'era una volta un re con sette guanciali che disse alla sua serva:
"Se avessi sette guanciali mi addormenterei e infatti ne ho proprio sette e quindi mi addormento."
e infatti si addormentò e qui finisce la storia del re con sette guanciali."
E qui finisce la storia del re con sei guanciali."
E qui finisce la storia del re con cinque guanciali.

```
PER SPIRALE :LATO
SE :LATO > 180 STOP
A :LATO
D 93
SPIRALE :LATO + 5
FINE
```

Osservando la storia (e la Traccia della spirale) si vedono facilmente alcune cose ormai ovvie, come:

- le ultime virgolette aperte sono le prime a venire chiuse e viceversa, cioè, prima che la storia dei cinque guanciali possa finire, deve finire quella dei sei, e così via (in Logo, prima che Spirale 40 possa finire deve finire Spirale 50).
- le procedure che non soddisfano il condizionale (Se ... Stop), per fermarsi arrivano a leggere la parola Fine che si trova dopo la chiamata ricorsiva.
- la storia e la procedura sono costruite (appositamente) in modo da avvicinarsi sempre di più alla condizione di fermata, che si trova dopo il Se. La condizione di fermata è chiamata di solito base della ricorsione, e l'azione che ci permette di avvicinarci ad essa (l'aggiungere ad ogni giro un guanciale), si chiama passo della ricorsione.

In questo caso potete semplicemente chiamare con un argomento sempre più piccolo una procedura come:

```
PER POLI :NUM :LATO
RIPETI :NUM [A :LATO D 360 / :NUM]
FINE
```

## Procedure trasparenti

Stanchi di triangoli e quadrati, supponiamo ora che qualcuno sfidi la nostra vena artistica chiedendoci di realizzare una fila di campanili in prospettiva (sempre più piccoli!).

Cominciamo ad affrontare il problema progettando una procedura per un campanile di altezza variabile. La nostra proposta è di utilizzare per il momento soltanto un rettangolo e un triangolo. Dopo avere disegnato il rettangolo, la tartaruga dovrà avanzare della lunghezza del lato e poi ruotare leggermente per fare il tetto. (Per definire più di una procedura alla volta, potete premere <Ctrl> C o <Run/Stop> soltanto alla fine dell'ultima procedura).

```
PER CAMPANILE :ALTEZZA
RETTANGOLO :ALTEZZA / 5 :ALTEZZA
A :ALTEZZA D 30
TRIANGOLO :ALTEZZA / 5
FINE

PER RETTANGOLO :BASE :ALT
RIPETI 2 [A :ALT D 90 A :BASE D 90]
FINE
```

```
PER TRIANGOLO :LATO
RIPETI 3 [A :LATO D 120]
FINE
```

Questo dovrebbe già funzionare. Ora, per disegnare la fila, potremmo usare una procedura che, dopo aver disegnato un campanile e spostato la tartaruga, chiama se stessa con un argomento più piccolo:

```
PER FILA :ALT
CAMPANILE :ALT
SPOSTATI 30
FILA :ALT - 10
FINE
```

```
PER SPOSTATI :LUNG
SU
D 90 A :LUNG S 90
GIU
FINE
```

Sembra ragionevole, eppure c'è qualcosa che non va: provare per credere (con Fila 80, ad esempio), i campanili più piccoli vengono tutti storti.

La rotazione di trenta gradi nella procedura Campanile, fa sì che la tartaruga inizi le varie chiamate con un orientamento sempre diverso. Un espediente semplice e incredibilmente utile per tenere tutto sotto controllo, è invece progettare le procedure in modo che la tartaruga torni sempre al suo stato iniziale. Il che vuol dire nello stesso punto sullo schermo e con il naso rivolto nella medesima direzione, come se niente fosse successo.

Chiameremo trasparente una procedura che non influisce sullo stato (posizione e orientamento) della tartaruga. Così, i nostri Rettangolo e Triangolo sono procedure trasparenti, e possiamo quindi usarle senza sorprese.

Osservate cosa provoca una rotazione totale di 360 gradi e uno spostamento uguale in ogni direzione: permette alla figura di chiudersi.

Tornando al nostro infelice caso, potremmo cercare di rendere trasparente la procedura Campanile. Visto che Rettangolo e Triangolo lo sono, sappiamo che nessuna delle due ci sposterà la tartaruga. Quindi, sarà sufficiente aggiungere in fondo a Campanile una linea di programma che ripercorra a rovescio gli altri spostamenti:

```
PER CAMPANILE :ALTEZZA
RETTANGOLO :ALTEZZA / 5
:ALTEZZA
A :ALT D 30
TRIANGOLO :ALT / 5
S 30 I :ALT
FINE
```

Ora rimane solo il problema di fermare la procedura Fila prima che arrivi a disegnare campanili negativi sempre più grandi:

```
PER FILA :ALT
SE :ALT < 20 STOP
CAMPANILE :ALT
SPOSTATI 30
FILA :ALT - 10
FINE
```

### Sempre più difficile

Le procedure ricorsive che abbiamo scritto finora avevano tutte la chiamata ricorsiva in fondo, ma non è certo obbligatorio fare così. In effetti c'era pur sempre qualcosa che esse facevano dopo la chiamata: si fermavano! Cioè, leggevano nel testo la parola Fine, prima della quale avrebbero potuto benissimo esserci altre istruzioni.

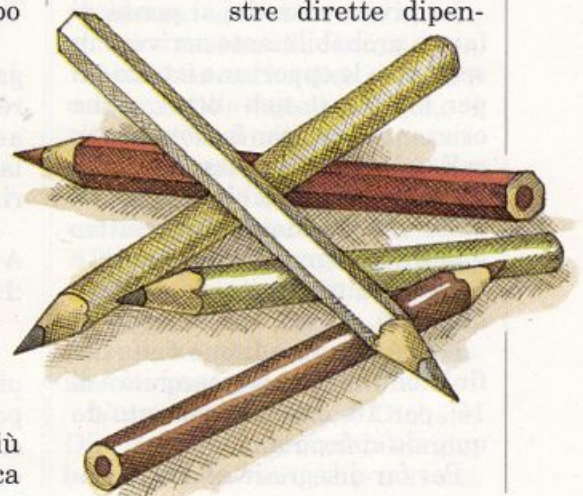
Sarebbe interessante rendere trasparente la procedura Fila: potremmo così disegnare qualcosa davanti al primo campanile, oppure una seconda fila sulla sinistra.

Sappiamo già che ogni copia della procedura Fila ne chiama un'altra con un parametro più piccolo, aspettandosi che finisca (proprio come una favola resta in sospeso quando al suo interno ne inizia un'altra). Se quindi ognuna di queste copie subito prima di terminare, pensa a far retrocedere la tartaruga di ciò che le spetta, alla fine il nostro triangolino dovrebbe ritrovarsi al punto di partenza:

```
PER FILA :ALT
SE :ALT < 20 STOP
CAMPANILE :ALT
SPOSTATI 30
FILA :ALT - 10
SPOSTATI - 30
FINE
```

Un modo abbastanza utile per comprendere il comportamento delle procedure è figurarsele come tanti omini, ognuno dei quali svolge alla lettera certe operazioni. Le primitive Logo (Avanti, Se, Stop, eccetera) sono un po' le api operaie di questo sistema, mentre le procedure che voi scrivete sono piccoli impresari che si limitano ad impartire ordini e ad aspettare che vengano eseguiti (potrebbero fare tutto per telefono). Se abbiamo diverse copie di una stessa procedura (come nel caso di chiamate ricorsive) pensiamo ad un omino diverso per ogni copia.

Seguiamo in questo modo l'esecuzione della chiamata Fila 30. Il nostro primo impresario (alle nostre dirette dipen-

denze) si segna sull'agenda il valore di :Alt (30) e chiede a Se se per caso 30 è minore di 20. Ottenuta risposta negativa, dopo aver chiamato Campanile 30 e Spostati 30, che già conosciamo bene, chiama Fila 20 e incrocia le braccia fino a che questa non avrà finito, dopodiché chiamerà Spostati 30 per tornare al punto di partenza.

Fila 20, nel frattempo, fa eseguire un campanile più piccolo e uno spostamento, e aspetta l'esecuzione di Fila 10 per ripercorrere lo spostamento a rovescio.

Fila 10, dal canto suo, verifica immediatamente (telefonando a Se) che 10 è minore di 20, e non fa altro che fermarsi allo Stop. Il controllo torna così a Fila 20, che dopo avere finito il suo lavoro (avere spostato indietro la tartaruga) lo passa a Fila 30, che finisce e lo passa direttamente a voi.

## Esercizi - Capitolo undicesimo

*1) Provate tutto quanto, e spiegate a qualcuno che vi sta vicino con meno di 250 parole la relazione tra la storia del re e la procedura:*

```
PER QUADRATO
A 40
D 90
QUADRATO
FINE
```

*nonché il funzionamento di:*

```
PER PASSEGGIA PER AZONZO
A ACASO 40 A 20
D 90 D 90 * ACASO 4
PASSEGGIA AZONZO
FINE FINE
```

*Come sempre, dopo la parola Fine, premere Ctrl C o Run/Stop. Mentre la tartaruga passeggia, usate i tasti F1, F3 e F5 per vedere la parte di schermo che vi interessa.*

*2) Osservate la Traccia delle procedure contenute nel secondo paragrafo. Se non conoscete la primitiva Traccia, consultate il paragrafo ad essa dedicato nell'appendice C. Osservate bene il modo in cui le varie chiamate di procedura vengono indentate (scalate di uno spazio) l'una rispetto all'altra, proprio come nella storia all'inizio del capitolo.*

*3) Scrivete altre procedure ricorsive che usino ACaso. Ricordate che, ad esempio, ACaso 50 riporta un numero compreso tra 0 e 49, ACaso 35 un numero tra 0 e 34, eccetera. ACaso 16 riporterà così un numero utilizzabile come colore (con ASCOL o CF).*

*4) Scrivete una procedura Logo sul modello di Storia che stampi la storia del re con sette guanciali (usando un parametro). Ricordatevi che per fondere assieme liste o parole (o entrambe) si può usare la primitiva Frase.*

### Alberi e scacchiere

A volte problemi che a prima vista appaiono complessi o noiosi hanno una soluzione ricorsiva sorprendentemente semplice. Per trovarla basta fare attenzione a certe regolarità che possono presentarsi ad ogni passo della realizzazione del disegno.

Per parlare più concretamente, voi come disegnereste una scacchiera?

La prima cosa che si pensa di fare è probabilmente scrivere in sequenza le opportune istruzioni per le nove (e non otto) righe orizzontali e per le nove verticali. Dopotutto possiamo considerare una scacchiera come un quadrato formato da quattro quadrati, ognuno dei quali è formato a sua volta da quattro quadrati...

E così comprendiamo nella definizione anche una scacchiera di 16 per 16, dipende soltanto da quando ci fermiamo.

Per far disegnare alla tartaruga tutto ciò, possiamo dirle di fare un quadrato che abbia in ogni angolo una scacchiera più piccola (col lato dimezzato), ovvero di ripetere quattro volte:

```
A :LATO D 90 SCACCHIERA
:LATO / 2
```

Poiché il lato diventa sempre più piccolo, possiamo utilizzarlo per la condizione di fermata, decidendo ad esempio che i quadretti della scacchiera non debbano misurare meno di dieci passi di tartaruga:

```
PER SCACCHIERA :LATO
SE :LATO < 10 STOP
RIPETI 4 [A :LATO D 90 SCAC-
CHIERA :LATO / 2]
FINE
```

Un altro dei classici della grafica ricorsiva è il disegno di un albero, forse perché con diagrammi simili (ma rovesciati) è possibile rappresentare graficamente il

*5) Disegnando poligoni (esagoni, per esempio) uno dentro l'altro si può creare l'illusione ditrovarsi dentro ad un tunnel: fatelo con unaprocedura ricorsiva. Non è necessario, ovviamente, che usiate una singola procedura. In questo caso potete semplicemente chiamare con un argomento sempre più piccolo una procedura come:*

```
PER POLI :LATO
RIPETI :NUM [A :LATO D 360 / :NUM]
FINE
```

*6) Introducete nella procedura Spostati un secondo parametro con la misura dell'angolo di rotazione. Potrete così creare l'illusione prospettica disegnando i campanili sempre più in alto.*

*7) Sbizzarritevi con file di case, fiori o cieli stellati.*

*8) Disegnate una ragnatela.*

*9) Non fa mai male ricordare l'esistenza della primitiva Traccia. Osservate attentamente Albero e Scacchiera con il suo aiuto.*

*10) Disegnate un abete, e poi una strada affiancata da abeti.*

*11) Ispirandovi a Scacchiera, progettate una procedura che disegni i quadrati più piccoli in fuori rispetto al quadrato principale. Fate lo stesso con altri poligoni, fino a disegnare un esagono che abbia su ogni angolo un pentagono, che abbia su ogni angolo un quadrato...*

*12) Scrivete una procedura ricorsiva che disegni un alveare (non è tanto facile, occorre lavorare di Sulapenna e Giulapenna).*

*13) Passate in rassegna le vostre vecchie procedure: ce n'è qualcuna in cui si può utilizzare Locale?*

*14) Come sappiamo, Leggering (abbreviato LR) riporta come risultato una lista con dentro la prima frase che viene digitata. Usatela insieme alla primitiva ACaso per un piccolo quiz che chieda le tabelline.*

funzionamento dei programmi.

Ci riferiamo ad un albero invernale, senza foglie, ma con rami che si biforcano sempre più frequentemente andando verso l'alto. Possiamo scomporre il problema in due: Albero penserà a disegnare il tronco, e Rami la biforcazione.

Incidentalmente, una biforcazione non è altro che un punto da cui si dipartono due Alberi più piccoli, orientati diversamente. Alla fine di ogni esecuzione di Albero la tartaruga tornerà al punto di partenza, grazie all'ultima istruzione:

```
PER ALBERO :ALT
SE :ALT < 3 STOP
A :ALT
RAMI :ALT / 2
I :ALT
FINE

PER RAMI :LUNG
S 25 ALBERO :LUNG
D 50 ALBERO :LUNG
S 25
FINE
```

Queste due procedure hanno una particolarità interessante: non sono ricorsive in senso stretto (non chiamano se stesse), ma si chiamano a vicenda, dando luogo ad un comportamento comunque ricorsivo.

## Impresari e cassetti

Il nostro impresario Fila di poco fa, prima di impartire gli ordini che gli competevano, di solito aveva cura di appuntarsi il valore del parametro :Alt indicato nell'intestazione (cioè di seguito al nome della procedura).

Questi parametri possono, come sapete, essere più d'uno (per esempio una semplice procedura come Rettangolo ne ha due) e ci si può immaginare che siano sistemati in una cassettiera della scrivania dell'impresario. Ogni cassetto avrà un nome (nel nostro caso Alt) e un singolo valore (20,30, eccetera).

Quando noi scriviamo :Alt (con i due punti) stiamo parlando di quello che c'è dentro al cassetto chiamato Alt, che sta nella scrivania dell'impresario. Nel caso di chiamate ricorsive, ci possiamo trovare con un sacco di impresari chiamati Fila, ognuno dei quali ha un cassetto Alt. Non è un problema: ognuno fa riferimento al suo e nessuno fa confusione.

Possiamo anche dire che Alt è una variabile Locale, cioè che il suo ambito è limitato ad una certa procedura; una volta terminata l'esecuzione della procedura la variabile viene buttata via.

Provate a chiedere Stampa :Alt dopo aver eseguito Fila, e vi verrà risposto che Manca il nome Alt, perché esisteva soltanto dentro alla procedura.

L'inverso di locale è Globale. Dire che una certa variabile è globale significa dire che non sta

nella scrivania di un particolare impresario, ma in una sorta di cassettiera principale a cui tutti gli impresari possono attingere.

Le variabili globali sono di solito quelle create con Assegna, ad esempio:

```
AS "ALT [QUESTA È UNA VARIABILE GLOBALE]
ST :ALT
```

Se ora fate eseguire di nuovo Fila, e poi controllate il valore di Alt, scoprirete che non è cambiato. Ogni chiamata della procedura Fila utilizzava il valore che le veniva passato come argomento.

Un altro modo di usare variabili locali è di dichiararle all'interno delle singole procedure usando la primitiva Locale. Con essa create un nuovo cassetto nella scrivania dell'impresario in questione, in modo che il successivo Assegna metterà il valore in quel cassetto invece che nella cassettiera principale:

```
PER DOMANDA
ST [10 SOLDATI SONO IN FILA]
ST [A 3 METRI L'UNO DALL'ALTRO.]
ST [QUANTO È LUNGA LA FILA?]
FINE

PER QUIZ
DOMANDA
LOCALE "RISPOSTA
ASSEGNA "RISPOSTA LEGGIRIG
SE :RISPOSTA = [27] ST [BRAVO!] STOP
SE :RISPOSTA = [30] ST [SBAGLIATISSIMO! SONO 27] STOP
ST [PENSACI MEGLIO:] QUIZ
FINE
```

Perché abbiamo usato Locale? In effetti la procedura avrebbe funzionato allo stesso modo anche senza, con la differenza che avrebbe distrutto il precedente eventuale valore di:Risposta. Noi saremmo perciò costretti a ricordarci di non mettere mai niente di importante in una variabile chiamata Risposta, perché correremmo il rischio di perderlo. Locale ci evita intanto questo tipo di preoccupazioni. Quando poi ci sono di mezzo chiamate ricorsive, le cose si complicano al punto di renderne quasi indispensabile l'uso.

## La prima parola di questa frase è la

Ma la prima parola di questa frase è questa.

La strana pratica Logo di usare virgolette e due punti a seconda delle occasioni, deriva dalla necessità di distinguere tra i vari tipi di oggetti che vogliamo via via indicare, in particolare tra la parola stessa ed il suo valore.

Quando diciamo che "Milano" ha sei lettere stiamo parlando di una parola, mentre se diciamo che Milano ha un milione di abitanti stiamo parlando di una città; nel primo caso abbiamo dovuto però usare le virgolette. Ovvero, l'oggetto indicato dalla parola "Milano" (quello che sta "dentro il cassetto", se ci riuscite) è una certa città della Lombardia, con tanto di tram e di ciminiere.

In Logo succede un po' la stessa cosa, con la complicazione che il valore di una parola può essere un'altra parola (o un numero, o una lista). La parola stessa è il nome del cassetto, mentre il suo valore è quello che c'è dentro.

Così, per assegnare un valore ad una parola dal fantasioso nome, Parola, dobbiamo digitare:

```
ASSEGNA "PAROLA qualchecosa
```

con le virgolette, perché stiamo parlano del nome del cassetto. Allo stesso modo:

```
LOCALE "PAROLA
```

si riferisce al cassetto chiamato Parola. Ancora:

```
ASSEGNA "PAROLA "PRECIPITEVOLISSIMEVOLMENTE
```

mette Precipitevolissimevolmente... nel cassetto Parola. E poi ancora:

```
ASSEGNA :PAROLA qualche cosa
```

(con i due punti) mette qualchecosa nel cassetto chiamato Precipitevolissimevolmente.

Torneremo sull'argomento delle parole e dei loro valori all'inizio del 13esimo capitolo.

**Fernando Zanini**

*(continua)*

*Si conclude su questo numero la trattazione del programma Extra Mind introdotto sul numero scorso e rivolto a chi ama apprendere, divertendosi, i concetti generali della programmazione basic e le potenzialità che lo caratterizzano.*

# Un buon esempio per imparare

Vediamo ora le restanti tre funzioni del programma, terminando il discorso con una sommaria descrizione del listato e delle principali procedure che lo costituiscono.

**Sconto commerciale**

È opportuna una premessa.

All'atto della sottoscrizione di un mutuo bancario, o comunque sia di un impegno monetario nei confronti di un istituto di credito, viene concordato un piano di ammortamento, o di rientro, redatto a cura dell'istituto stesso, con criteri che in linea di massima hanno carattere omogeneo per quasi tutte le strutture del genere operanti in Italia. Tutti i dati che confluiscono alla stesura di tale documento hanno, o dovrebbero avere, una analogia oggettiva. Il capitale iniziale, o erogato, e la durata dell'operazione sono dati senza storia e senza commenti. Il tasso d'interesse è oggetto di negoziazione tra le parti, e più che altro è frutto di potere contrattuale.

Il quarto dato che contribuisce alla quantificazione del debito totale (montante), ed alla conseguente determinazione delle rate sono i giorni banca. Questo dato viene espresso da una cifra che è la risultante di una sommatoria. È inutile, in questa sede, analizzare dettagliatamente i criteri con cui la stessa viene costruita; basti dire che o è nota (ma gli istituti non la comunicano mai), o può essere costruita nel corso del programma da una apposita subroutine.

Dopo avere premuto la lettera S lo schermo offre la possibilità di avere delle istruzioni, che in realtà (anche se più succintamente) riflettono quanto indicato nel presente capitolo. Se la risposta è negativa, si evidenzia un submenù che propone quattro ipotesi: debito finale (montante); giorni banca; numero rate; rate o interesse. Non tragga in inganno la voce giorni banca; a questo livello l'uso di questa routine serve solo se sono noti capitale iniziale, montante - indicato nel piano d'ammortamento elaborato dall'istituto - numero delle rate e tasso d'interesse), a stabilire quale numero di giorni banca è stato utilizzato. In concreto, a questo livello del programma, ognuno dei dati che dovesse essere calcolato deve esserlo in funzione della conoscenza degli altri tre.

A questo punto, qualsiasi opzione venga prescelta (salvo giorni banca, sarebbe un nonsenso per quanto segue) offre come prima schermata, la proposta di effet-

tuare il calcolo di tale dato. Le risposte possibili sono tre (salvo la solita uscita al menù principale): sì (y); no (n) e submenù (nel caso ci si fosse accorti di aver sbagliata l'opzione).

Nel caso di risposta negativa, lo schermo successivo pone le domande per la soluzione del problema desiderato ed elabora i dati proposti ad esaurimento del proprio compito.

In caso di risposta affermativa, la prima successiva schermata richiede la precisazione del tipo di rateazione prevista (da indicarsi come sempre con la sola pressione della lettera iniziale); fornito tale dato, vengono richiesti tre ulteriori dati che richiedono una piccola illustrazione.

• **Durata (anni):** questo dato è intuitivo e necessita della sola indicazione del numero degli anni per cui è prevista l'operazione;

• **Decorrenza** (mesi): è possibile, in talune circostanze, concordare con gli istituti l'inizio del rientro con un certo differimento: ad esempio dal nono mese dall'erogazione. In tale caso dovrà essere scritto 9. Se, come più usuale, la decorrenza è immediata, dovrà essere indicato 1. In ogni caso dovrà essere fornito il dato corretto, che può assumerla tutta in una, ma una sola gamma di valori.

• **Giorni incasso**: nel compito dei giorni banca, gli istituti assumono che,contro la puntuale corresponsione delle rate, da parte loro, per il materiale possesso del pagato, possano intercorrere un certo numero di giorni. Diciamo che un valore medio è 10. Poiché anche questo è un aspetto che può essere negoziato, dovrà essere indicato il numero frutto dell'accordo.

A questo punto, in calce e lasciando tutti i dati in evidenza, vengono quantificati giorni banca e numero delle rate. Tra parentesi viene indicato che nella successiva elaborazione, essendo questi dati memorizzati, non sarà più necessario trascriverli alla rispettiva voce, essendo sufficiente premere Return. Questo è un aspetto di comodità: nulla cambia inserendoli da tastiera.

Lo schermo successivo propone la selezione effettuata all'inizio; basterà inserire gli ulteriori dati e verrà fornita la risposta desiderata.

Corre l'obbligo di informare che (per ragioni misteriose) i giorni banca utilizzati dagli istituti sono di norma leggermente superiori a quelli elaborati (peraltro in maniera corretta) da questo programma. Se ne potrebbe trarre la conclusione che non per tutti due più due fa quattro. Le differenze sono comunque di ordine estremamente modesto.

Al termine, come sempre, compaiono le solite opzioni per la ripetizione o per l'uscita.

## Tempi orari

La storia di questo programma è articolata nel tempo. È stato costruito a misura delle necessità ,o quantomeno dal gusto di verificare certe situazioni ,e per risolvere problemi di curiosità.

Quanti non si sono divertiti, in autostrada, su una base ragionevolmente misurata, a controllare in quanto tempo veniva percorso un chilometro? Certo che poi occorre trasformare questo tempo in una media. Oppure, chi non ha cercato di prendere il tempo sul giro nel corso di una gara di formula 1? L'impresa è impossibile, considerato che viene rilevato il centesimo di secondo, ma se qualcuno, diciamo il commentatore televisivo, fornisce il dato, può essere curioso trasformarselo con mezzi propri in una media, magari verificare quanto viene poi comunicato, sapere quanto tempo si impiegherà, ad una certa media (ingorghi a parte) a percorrere un certo numero di chilometri. Così è nato di giorno in giorno, questo programma che illustreremo subroutine per subroutine.

Premendo il tasto T, si presenta la domanda se siano necessarie le istruzioni. Come per lo sconto commerciale, la risposta positiva attiva due schermi che illustrano, molto sommariamente, quanto in questa sede cercheremo di spiegare più dettagliatamente. Al termine, o in caso di risposta negativa, compare il menù della sezione tempi orari, che è l'unico le cui opzioni vengano pilotate da numeri.

**1 - media su kilometro:** alla domanda dello schermo occorre semplicemente rispondere inserendo il numero dei secondi, digitare il punto (.) e aggiungere le frazioni nella misura che riterrete di aver rilevato. Si consideri che il computer elabora un numero elevatissimo di decimali, per cui anche fornirgli i centesimi od i millesimi di secondo, per esso non costituisce problema, né in termini di precisione di calcolo né di tempo di elaborazione. Viene visualizzata la media oraria (sul chilometro) ed alla prosecuzione la solita richiesta circa la ripetizione, le uscite al submenù o al menù principale.

**2 - media su percorrenze:** in linea di massima questa routine si riallaccia alla precedente, con una diversità: qui deve essere indicato il numero dei chilometri, oltre al tempo di percorrenza.

Prima del punto vanno digitati detti chilometri, subito dopo le centinaia, le decine ed i metri. Anche per quanto riguarda il tempo, a sinistra del punto le ore, a destra i minuti (2 cifre), i secondi (2 cifre) e, sempre tutto di seguito, le frazioni. A titolo di esempio, un percorso di 176 chilometri e 987

metri, percorso in un'ora, 32 minuti, 26 secondi e 187 millesimi, verrebbe indicato come segue:

chilometri 176.987
tempo 1.3226187

Qualora il tempo fosse di due ore, zero minuti e 35 secondi, la grafia corretta sarà: 2.0035. Da ciò si desume che gli spazi intermedi, anche se nulli, debbono essere riempiti con degli zeri. Alla prosecuzione compaiono le solite opzioni circa la ripetizione o le diverse uscite.

**3 - loop percorsi brevi:** questa è la routine per rilevare le medie su percorsi (brevi, o percorsi da mezzi veloci) di cui sia nota la lunghezza. La prima domanda che pone lo schermo è proprio tale lunghezza, sempre con a sinistra del punto i chilometri, ed alla destra il resto.Alla successiva domanda dovrà essere digitato il tempo; trattandosi di percorsi brevi, a sinistra del punto dovranno indicarsi i minuti, alla destra i secondi e frazioni, senza soluzione di continuità, ma sempre con l'accortezza che zero secondi dovranno essere indicati col doppio zero. Una volta fornita la risposta, il computer si predispone a ricevere il nuovo dato di tempo, e così via sino a che, come indicato nella maschera, non digiterete lo 0 seguito da Return. È chiaro che lo zero, seguito da punto e cifre predispone la macchina ad un conteggio, per cui non esiste il pericolo che il programma possa interrompersi per tempi inferiori al minuto. L'interruzione attraverso lo zero propone la solita casistica di eventuale ripetizione o accesso ai menù.

**4 - loop su percorrenze:** analoga alla precedente routine, se ne discosta per il fatto che i percorsi possono essere di qualsiasi lunghezza e pertanto i tempi che si prevedono, alla sinistra del punto, sono le ore, ed alla destra, in serie come indicato ai punti precedenti, tutti gli altri valori. Anche in questo caso, un tempo inferiore all'ora, e che pertanto inizia con lo zero, non crea pregiudizio al calcolo ed alla sua iterazione. Analoghe anche le considerazioni circa l'uscita.

**5 - percorsi da media:** la storia di questa routine è curiosa. Volevamo conoscere la lunghezza dell'anello dell'autodromo di Monza, e sapevamo che qualcuno lo aveva percorso in un minuto, 26 secondi e 16 centesimi, alla media di km. 242.340. Abbiamo costruito l'algoritmo, realizzata la routine e abbiamo saputo che l'anello di Monza è di 5 chilometri ed 800metri: anzi di km 5,800004.

La prima domanda che rivolge lo schermo si riferisce al tempo impiegato (a sinistra del punto ci sono i minuti); quindi la media realizzata, verrà visualizzata sul percorso compiuto. Anche in questo caso compaiono le solite opzioni per l'uscita.

**6 - :** in quanto tempo dovete percorrere un chilometro per non superare la media dei 110 km/h fissata per legge? In 32 secondi e 727 millesimi. Lo conferma questa routine, che in apertura, chiede il numero dei chilometri (da scrivere alla sinistra del punto) e che segue con la richiesta della media che si intende o si intenderebbe tenere. Lo schermo mostra, ben evidenziati e separati, ore, minuti, secondi e millesimi di secondo. Anche in questo caso compaiono le solite opzioni per l'uscita.

## Valore attuale

Questa routine è esattamente il contrario dell'interesse composto. Si pensi di dover incassare una certa cifra tra un determinato numero di anni. Ad esempio una polizza sulla vita di cui si sospende il pagamento, acquisendo il diritto, sia pure ridotto, di ricevere un certo capitale, ma solo alla scadenza naturale del contratto. È possibile disporre immediatamente di detto ammontare, ma se ne calcola il valore attuale, ossia si deve determinare quale sia l'importo che, capitalizzato per gli anni mancanti ad un certo tasso (nel caso delle assicurazioni sulla vita, è o era, il 4.75%), fornisca il valore finale o montante.

Pertanto in questa routine le domande che pone lo schermo sono:

- montante in luogo di capitale;
- differimento in luogo di durata;
- interesse come nell'interesse composto.

Una volta inseriti, questi dati nell'ordine richiesto, viene fornito il capitale iniziale. Uscendo dalla routine, ed utilizzando quella dell'interesse composto, sarà facile verificare l'esattezza del conteggio.

Si fa presente che potranno riscontrarsi minime differenze, nell'ordine delle lire, dovute al fatto che, operando il computer attraverso i logaritmi, non sempre le riconversioni di andata hanno un riscontro esatto con quelle di ritorno.

Si consideri inoltre che anche in questo caso la durata, o differimento, può essere indicata con valori non interi. Come già detto, le frazioni di anno vanno espresse non col numero dei mesi, ma decimalizzando tale dato. Non è un conto particolarmente complicato: basta dividere per 12. Mesi 6 diviso 12 = 0.5. Mesi 3 diviso 12 = 0.25, e così via. Si può incorrere in numeri periodici (i due mesi sono 0.1666...). Basta indicare le prime tre cifre decimali. E, ovviamente, premettere il numero degli anni.

**Giuseppe Alberti Monzoni**

## SAPERE COSA C'È

Tutti i prodotti per l'ufficio, tutto l'hardware e tutto il software, tutte le soluzioni del mondo: 140.000 mq, 850 espositori in rappresentanza di 1800 Case; più di 300 novità; 13 convegni internazionali e specializzati.

## SAPERE COSA C'ERA

Seconda tappa della Mostra-telecomunicazioni: è dedicata ai dati e alle immagini. Patrocinata dalla Fiera Milano e da Anie, è organizzata da Smau e Sip con la partecipazione di Alcatel Face, Fatme-Ericsson, Italtel, Siemens Telecomunicazioni, Sirti, Telettra.

## SAPERE COME ORIENTARSI

Un'assoluta novità: il sistema "Gulliver", una guida elettronica di nuova concezione che vi conduce alle soluzioni che cercate, dicendovi il cosa, il dove, il quando, il chi, il come e il perché.

## SAPERE QUALE DESIGN

Patrocinata dal Comune di Milano, all'Arengario di piazza Duomo c'è la mostra "La forma del lavoro - Venti anni di Premio Smau Industrial Design".

**SAPERE QUANDO E DOVE ANDARE**

**Smau: 5-9 Ottobre, Quartiere Fiera Milano (comodissimo il Metrò: la fermata Amendola-Fiera è di fronte a uno degli ingressi Smau). Mostra "La forma del lavoro - Venti anni di Premio Smau Industrial Design": 21/9-15/10, Arengario, in Piazza Duomo.**

**Segnaliamo infine Eimu, Esposizione Internazionale Mobili Ufficio, 20-25 Settembre, Quartiere Fiera Milano.**

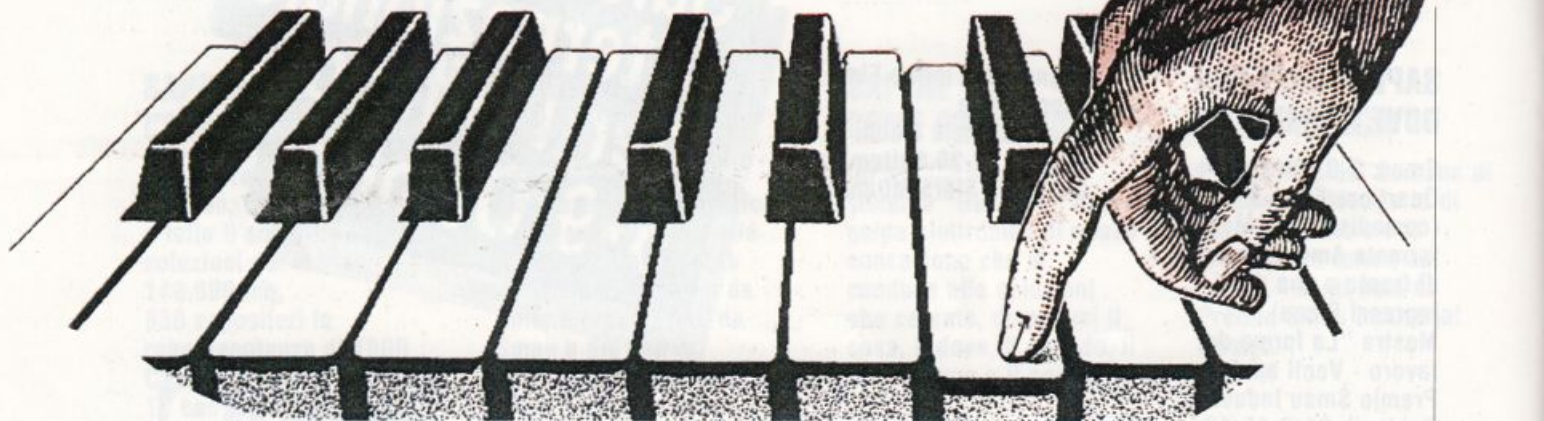
# Dimmi di SID!

*Tuttora l'interfaccia Sid (Sound Interface Device) è tra le più potenti e versatili nel mondo degli home computers, nonostante la veneranda età del nostro Commodore 64. Programmare in modo fruttuoso questo validissimo chip, però, è spesso arduo, soprattutto perché manca una sufficiente documentazione ufficiale.*

Chiunque abbia tentato di creare effetti sonori complessi, per poi riutilizzarli nei propri programmi, si sarà certamente accorto che non è così semplice programmare il chip Sid. Oltre a questo gli effetti ottenuti non sempre permettono l'elaborazione di un eventuale programma principale a cui la musica o l'effetto sonoro devono fare da sottofondo, in quanto è molto complicato sfruttare completamente tutte le potenzialità del Sid senza interferire con la velocità di esecuzione dei nostri programmi, anche in linguaggio macchina.

Come se non bastasse, la maggior parte delle potenzialità del Sid sono sconosciute o inutilizzate, come i filtri, la risonanza, il sincronismo e la modulazione ad anello. La difficoltà nel programmare queste funzioni sta anche nel fatto che, pur conoscendo le locazioni di memoria in cui lavorano, non si conosce il significato e l'effetto, nonché le relazioni che le legano.

Nelle puntate che seguiranno, oltre a utilissimi programmi di gestione del Sid, saranno ampiamente documentate tutte le sue più ignote caratteristiche, corredate da esempi semplici ma attentamente dettagliati.

Il chip Sid è composto da tre canali programmabili separatamente, che hanno la possibilità di interagire gli uni con gli

altri tramite il sincronismo e la modulazione ad anello.

Il programma di questo mese permette di utilizzare in maniera semplicissima tutte le potenzialità del Sid. Inoltre sono presenti due nuovi comandi per ottenere grandiosi effetti speciali.

## Note per il caricamento

Digitare LOAD"SID" e premere il tasto Return.

Al termine del caricamento potete dare il consueto Run; lo schermo si cancellerà e apparirà la schermata di presentazione con due possibilità di scelta:

**• Il tasto con il segno di sottrazione (-).**

Permette di visualizzare la tabella valori dell'inviluppo, che è composta da tre fasi distinte: Attacco, Caduta, Sostegno e Stacco.

Linviluppo costituisce la temporizzazione della nota in modo totalmente indipendente dalla frequenza, può essere paragonato a un volume secondario gestito interamente da noi.

Per chiarire meglio le idee supponiamo di aver impostato un determinato effetto sonoro, e di aver selezionato il volume al massimo (15). In questo momento però dovete avere tutti i parametri dell'inviluppo azzerati: (Attacco=0 ; Caduta=0 ; Sostegno=0 ; Stacco=0); se ora accendete il gate della voce da voi utilizzata, il chip Sid non emetterà alcun rumore nonostante il volume sia al massimo; questo è dovuto essenzialmente a quella parte dell'inviluppo detta Sostegno, che a differenza delle altre tre si misura in percentuale (%), e non in maniera temporale (sec.).

Se ora avete sotto gli occhi la tabella valori presente nel programma, avrete anche notato che il Sostegno è l'unica parte dell'inviluppo che ha una gamma di valori pressoché completa (0-100%), tuttora selezionata a zero e, essendo in funzione del volume principale non bisogna stupirsi se il computer non dà alcun segnale, in quanto tutti sanno che lo 0% di 15 dà come risultato zero.

Se ora voi impostate il Sostegno ad esempio a 60% (9), il volume di uscita sarà il 60% di 15, cioè 9. Questa volta, acceso il gate, udrete il vostro effetto sonoro abbastanza nitidamente e potrete variarne il volume lavorando semplicemente sul Sostegno, come se fosse un volume secondario.

### Tavola 1.

**Elenco dei comandi del programma Sid**

- ***Tasto F1*** *Abilita la voce 1*
- ***Tasto F3*** *Abilita la voce 2*
- ***Tasto F5*** *Abilita la voce 3*

*STANDARD PER I CANALI 1-2-3:*
- ***Tasto A*** *Attacco*
- ***Tasto D*** *Caduta*
- ***Tasto S*** *Sostegno*
- ***Tasto R*** *Stacco*
- ***Tasto F*** *Frequenza oscillatore*
- ***Tasto P*** *Pulsazione onda quadra*
- ***Tasto W*** *Tipo di onda sonora*
- ***Tasto I*** *Modulazione ad anello*
- ***Tasto Y*** *Syncronismo*
- ***Tasto G*** *Interruttore voce*

*FUNZIONI SPECIALI CANALE 3:*
- ***Tasto O*** *Oscillatore accoppiato*
- ***Tasto E*** *Inviluppo accoppiato*
- ***Tasto B*** *Disabilita il canale 3*

*COMANDI D' USCITA:*
- ***Tasto Q*** *Frequenza filtro*
- ***Tasto 1*** *Filtra canale 1*
- ***Tasto 2*** *Filtra canale 2*
- ***Tasto 3*** *Filtra canale 3*
- ***Tasto M*** *Tipo di filtro*
- ***Tasto R*** *Risonanza filtro*
- ***Tasto V*** *Volume*

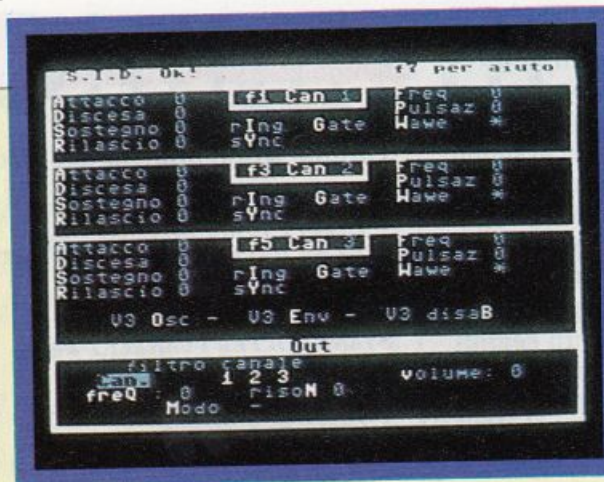
Ovviamente se il volume principale fosse stato 11 invece che 15, pur selezionando lo stesso valore di sostegno 60% (9), il volume d'uscita sarebbe stato il 60% di 11, che è circa 6,5.

Ora voi vi chiederete l'utilità di questo doppio volume, poiché, oltre a rendere un po' più preciso il volume d'uscita, non ha altre peculiarità. Niente di più sbagliato! Il volume principale oltre a influenzare il Sostegno, ha contemporaneamente un effetto sugli altri tre parametri. Infatti, una volta acceso il gate, il suono arriva istantaneamente al volume fissato, e questo può risultare scomodo ad esempio in una melodia, dove si richiede un'uscita dolce e continua delle note, e non a scatti.

L'inconveniente può essere aggirato selezionando l'Attacco a 3 secondi (13), e la Caduta a 2.4 secondi (11).

Ora se accendete il gate, il suono arriverà dolcemente al volume principale (Attacco) per poi ricadere sempre dolcemente al valore di sostegno (volume secondario), dato in percentuale dal volume principale.

Selezionando il valore 3 (secondi) per l'Attacco, abbiamo imposto al Sid di impiegare 3 secondi per arrivare da zero al volume massimo. In seguito, impiega al-

*La tabella dei valori dell'inviluppo. Nella terza colonna, espressi in percentuale, i valori di volume per ogni valore di sustain (a sinistra della tabella).*

tri 2.4 secondi per raggiungere il valore di Sostegno (Caduta). Questo vuol dire che il suono impiega 2.4 secondi + 3 secondi = 5.4 secondi per arrivare al volume di sostegno, passando per il volume principale.

Fino a che il gate rimane acceso il suono continua persistentemente a sentirsi ma, una volta spento, il suono passa bruscamente al valore zero, in quanto il valore contenuto nel registro di Stacco è 0. Se invece noi impostiamo lo Stacco a 9 secondi (13), sempre riferendoci alla tabella valori dell'inviluppo, una volta spento il gate il suono impiegherà esattamente 9 secondi per raggiungere il valore 0.

Ecco un esempio completo di tutto ciò che è stato spiegato finora:

- impostare il volume a 13,
- selezionare l'onda a dente di sega della voce 1 (S),
- definire la frequenza, sempre della voce 1, attorno ai 120 Hz (2000),
- impostare l'inviluppo come segue: Attacco 0.8 sec. (11), Caduta 0.75 sec. (9), Sostegno 93% (14) = 12.1, Caduta 3 sec. (12),
- premere il tasto funzione F6 (On/Off). Il suono raggiunge dolcemente il valore di sostegno, e sempre dolcemente torna a 0.

Le variazioni temporali che si possono ottenere con l'inviluppo sono pressoché infinite e sono modificabili comodamente a seconda delle esigenze.

Vediamo ora di generare un effetto che sfrutti adeguatamente l'inviluppo, e che sia allo stesso tempo interessante:

- impostare il volume al massimo (15)
- portare a 15 il Sostegno della voce 1
- selezionare la modulazione ad anello e la forma d'onda triangolare sempre della voce 1
- impostare la frequenza dell'oscillatore 3 a 6000 (360Hz) e impostare la disabilitazione
- definire i valori dell'inviluppo della voce 3 come segue: Attacco 13, Caduta 13, Sostegno 13, Stacco 12.
- impostare il gate della voce uno e lasciarlo sempre fisso.

Dovrebbe sentirsi un suono abbastanza monotono, ed è proprio ora che dovete lavorare sul gate della voce 3, accendendolo e spegnendolo a vostro piacimento.

L'effetto che si otterrà sarà piuttosto particolare, e varierà notevolmente a seconda dei parametri di inviluppo che inserirete.

**• Il tasto con il segno di addizione (+).**

Consente di accedere al cuore del programma. Una volta entrati in editazione, premendo il tasto funzione F7 apparirà un menù mobile che elenca i vari metodi per definire i parametri del Sid:

- *Il tasto F2:* equivale ad un interruttore generale (On/Off). Se viene premuto una volta accenderà tutti i canali del Sid, se viene premuto una seconda volta li spegnerà.
- *Il tasto F6:* permette di accendere e spegnere a breve distanza tutti e tre canali del Sid; risulta utile quando si intende verificare l'inviluppo di un effetto sonoro.
- *Il tasto <Shift> + <Return>:* permette di incrementare i valori rapidamente.
- *Il tasto <Return>:* consente di decrementare i valori sempre rapidamente.
- *Il tasto <Shift> + <Crsr L/R>:* permette di incrementare i valori a velocità moderata.
- *Il tasto <Crsr L/R>:* consente di decrementare i valori a velocità sempre moderata.
- *Il tasto <Shift> + <Crsr U/D>:* permette un incremento pari a un' unità alla volta.
- *Il tasto <Crsr U/D>:* consente invece di decrementare i valori sempre singolarmente.

Ora che abbiamo visto in che modo impostare i valori cerchiamo di utilizzarli. Lo schermo appare diviso in quattro zone, di cui le prime tre fanno riferimento ai registri standard di ogni singolo canale del Sid; la quarta zona invece delimita tutti quei comandi che influenzano il campione sonoro d'uscita: (Filtri, Risonanza, Volume).

Sono presenti inoltre tre comandi speciali che riguardano il canale 3.

## Tutti i comandi

In **tavola 1** trovate un elenco completo dei comandi, che ora documentiamo dettagliatamente.

• **Tasto A (attacco):** definisce il tempo che impiega l'effetto sonoro a raggiungere il volume massimo (0-15).

• **Tasto D (decay):** definisce il tempo che impiega l'effetto sonoro per raggiungere il valore di sostegno (0-15).

• **Tasto S (sustain):** seleziona il valore di sostegno espresso in percentuale del volume massimo (0-15/0-100%).

• **Tasto R (release):** definisce il tempo che impiega l'effetto per tornare a volume nullo (0-15).

Queste 4 opzioni costituiscono l'inviluppo dell'onda e si riferiscono alla tabella valori iniziale.

• **Tasto F (frequenza):** seleziona la frequenza dell'oscillatore in un campo da 0 a 4 KHz. Per ottenere il valore reale della frequenza basta moltiplicare il valore che appare sullo schermo (0-65535) per una costante pari a 0,0596.

• **Tasto W (wave):** determina la forma d'onda dell'oscillatore: T=triangolare, S=dente di sega, P=onda pulsante, N=rumore (onda disordinata).

• **Tasto P (pulsation):** questa opzione ha senso solo se usata in concomitanza con la forma d'onda pulsante, e permette di selezionare il tipo di onda quadra generata dall'oscillatore. Varia da circa 0,5% a circa 99,5%. Nel primo caso gli impulsi d'onda saranno strettissimi, nel secondo invece, molto larghi. Ovviamente selezionando un valore pari a 50% si ottiene un'onda quadra perfetta.

• **Tasto I (ring):** seleziona la modulazione ad anello, che per ovvi motivi di comprensione da parte del lettore, non possiamo documentare accuratamente. Possiamo solo dire che una volta impostata si seleziona automaticamente la forma d'onda triangolare della voce 1, e tramite una combinazione degli oscillatori 1 e 3 si ottiene che, variando la frequenza della voce 3 si induce una variazione della forma d'onda della voce 1. Per quanto riguarda il canale 2 la combinazione avviene tra gli oscillatori 1 e 2, mentre per il canale 3 risulta essere tre, gli oscillatori 3 e 2.

• **Tasto Y (sincronismo):** questo comando è analogo al precedente. Una volta che esso è stato selezionato, la voce 3 si sincronizzerà con la voce 1 rendendo le due voci il più contigue possibile, quasi da sembrare una singola voce.

• **Tasto G (gate):** equivale a un vero e proprio interruttore della voce in questione.

Passiamo ora alle funzioni speciali della voce 3:

• **Tasto O (oscillatore 3):** una volta premuto, tramite una routine in linguaggio macchina che legge la forma d'onda dell'oscillatore 3, si possono ottenere vari effetti speciali. Questo comando ha tre differenti caratteristiche (L,H,F) che hanno influenza sulla routine in esecuzione.

• **Tasto E (envelope):** è analogo al precedente ma questa volta i valori letti dalla routine sono quelli dell'inviluppo, sempre riferiti all'oscillatore 3.

• **Tasto B (disabilita voce 3):** questo comando si spiega da solo e torna utile soprattutto se usato assieme agli altri due, evitando noiosi rumori che non interessano direttamente l'operatore.

Passiamo ora alle opzioni di uscita che sono essenzialmente due, filtri e volume:

• **Tasto Q (freq. filtro):** seleziona la frequenza del filtro di uscita in un campo da 0 a 12 KHz. Per ottenere il valore reale della frequenza basta moltiplicare il valore sullo schermo, per una costante pari a 40,95.

| | Attack (sec) | Decay (sec) | Sustain (%) | Release (sec) |
|---|---|---|---|---|
| 0 | 0.002 | 0.006 | 0 | 0.006 |
| 1 | 0.008 | 0.024 | 7 | 0.024 |
| 2 | 0.016 | 0.048 | 13 | 0.048 |
| 3 | 0.024 | 0.072 | 20 | 0.072 |
| 4 | 0.038 | 0.114 | 27 | 0.114 |
| 5 | 0.056 | 0.168 | 33 | 0.168 |
| 6 | 0.068 | 0.204 | 40 | 0.204 |
| 7 | 0.08 | 0.24 | 47 | 0.24 |
| 8 | 0.1 | 0.3 | 53 | 0.3 |
| 9 | 0.25 | 0.75 | 60 | 0.75 |
| 10 | 0.5 | 1.5 | 67 | 1.5 |
| 11 | 0.8 | 2.4 | 73 | 2.4 |
| 12 | 1 | 3 | 80 | 3 |
| 13 | 3 | 9 | 87 | 9 |
| 14 | 5 | 15 | 93 | 15 |
| 15 | 8 | 24 | 100 | 24 |

*Il ricco pannello di controllo del programma. I primi tre settori dall'alto gestiscono ognuna delle tre voci del Sid.*

• **Tasto M (mode):** imposta il tipo di filtro che si intende utilizzare tra tre differenti combinazioni: Passa Basso (LP), Passa Banda (BP), Passa Alto (HP). La prima combinazione permette il passaggio delle frequenze inferiori a quella selezionata per il filtro, la seconda permette invece il passaggio di una gamma di frequenze attorno al valore impostato, infine la terza lascierà passare solo le frequenze superiori a quella impostata.

• **Tasti 1-2-3 (canali):** selezionano il filtro per ciascuno dei tre canali, e possono essere usati contemporaneamente.

• **Tasto R (resonance):** imposta la risonanza del filtro, che opera attorno al valore di chiusura (0-15).

• **Tasto V (volume):** determina il volume d'uscita dell'effetto sonoro; è prioritario rispetto agli altri comandi, infatti se viene posto a zero, qualunque cosa si selezioni, il Sid non emetterà alcun segnale; varia da 0 a 15.

## Effetti

Vediamo ora alcuni effetti che tutti possono sperimentare:

• **Treno a vapore.**
- impostare il volume al massimo: (15)
- impostare i valori di sostegno della voce 1 e 3 al massimo: (15)
- impostare il rumore come forma d'onda dell'oscillatore 1: (N)
- impostare l'onda a dente di sega dell'oscillatore 3: (S)
- impostare la frequenza dell'oscillatore 3 a (15)
- premere due volte il tasto (O), in modo che appaia di fianco alla scritta V3 Osc. la lettera H.
- ora accendete tramite il tasto (G) le voci 1 e 3.

Il suono che udrete sarà simile a quello di un treno a vapore. Aumentando la frequenza della voce 3 il treno aumenterà la sua andatura. L'effetto è più realistico procedendo ad alta velocità.

• **Gorgoglio elettronico.**
Un altro effetto. Premete il tasto F2 per azzerare il Sid e poi impostate i seguenti parametri:
- impostare il volume e il sostegno delle voci 1 e 3 al massimo: (15)
- selezionare l'onda triangolare dell'oscillatore 1: (T)
- selezionare l'onda a dente di sega dell'oscillatore 3: (S)
- impostare la frequenza sempre dell'oscillatore 3 a (400)
- disabilitare il canale 3: (B)
- premere due volte la lettera O: (V3 Osc. =H)
- accendere le due voci (1-3) con il tasto (G)

Il suono che si ottiene è già buono, ma provate ad aumentare la frequenza della voce 3 e sentirete che cambiamento!

• **Passi.**

Voce 1: - Sustain (15)
- Onda (N)

Voce 3: - Sustain (15)
- Onda (S)
- V3 Osc = (H)
- Disab. voce 3
- Frequenza (40)

Uscita: - Filtra voce (1)
- Freq. filtro (600)
- Modo filtro (LP)
- Volume (15)

Ora accendete i gate (G) rispettivamente della voce 1 e 3. L'effetto creato sembra quasi reale ed è possibile modificarlo completamente.

Gli effetti sonori che si possono inventare sono infiniti e, grazie ad un programma, che verrà pubblicato sul prossimo numero della rivista, saranno anche trasformati in routine in linguaggio macchina completamente rilocabili e occuperanno pochissima memoria. Grazie all'interrupt, potranno essere attivati da un programma Basic (o linguaggio macchina), mentre questo continua indisturbato la sua elaborazione ( per esempio, la visualizzazione del suono riprodotto).

Un ultimo consiglio prima di augurarvi buon lavoro: ogni tanto può capitare, soprattutto se usate molti comandi contemporaneamente, che l'elaborazione del suono si arresti, ma non preoccupatevi, basta premere il tasto asterisco (*), perché tutto torni alla normalità.

Per ora sbizzarritevi con questo programma, e quando pensate di aver trovato l'effetto sonoro che fa per voi, scrivete su un pezzo di carta i valori: vi torneranno sicuramente utili con il prossimo programma.

**Franco Valter**

# Exec: librerie e dispositivi

*In questo nostro ultimo appuntamento tratteremo il sistema delle librerie e quello dei dispositivi e vedremo la loro implementazione nell'Amiga integrando il tutto con alcuni esempi di gestione di questi sistemi in C.*

Essenzialmente una libreria è una collezione di funzioni che possono essere utilizzate, ovvero chiamate come subroutine, da un generico programma.

Esempio tipico di libreria sono le cosìdette librerie di compilazione, da dove vengono estratte le diverse routine da collegare al programma principale, affinché questi possa utilizzarle.

Durante la stesura di un programma in C, le diverse procedure presenti all'interno dello stesso programma, potranno chiamare delle funzioni standard precompilate quali ad esempio Printf o Scanf.

Queste funzioni non saranno presenti all'interno del segmento compilato, ma verranno aggiunte, in un secondo tempo, dal linkatore (Link). Per l'esattezza, il linkatore per prima cosa cercherà nei diversi moduli oggetto dei riferimenti irrisolti, ovvero delle chiamate a subroutine non contenute nei vari moduli oggetto, successivamente scandirà le diverse librerie di compilazione cercandovi le sopradette funzioni, che una volta trovate, verranno incluse nel codice eseguibile generato in uscita.

I principali vantaggi offerti dall'uso delle librerie di compilazione derivano dal fatto che le diverse funzioni in esse contenute verranno compilate una volta per tutte, e verranno aggiunte al programma principale in fase di linkatura, facendo risparmiare il

tempo che si sarebbe impiegato se queste funzioni, inserite nella sorgente del programma principale, fossero state ricompilate ogni volta.

Una seconda caratteristica positiva delle librerie di compilazione deriva dal fatto che nel codice eseguibile verranno inserite solo quelle funzioni effettivamente chiamate e non tutto il corpo della libreria.

### Librerie run-time

In un sistema multitask, diversi programmi potranno essere attivi contemporaneamente, chiamando le varie subroutine preposte a diversi compiti.

Ora accade spesso che una certa parte di queste funzioni siano in comune, ovvero, che diversi programmi utilizzino una stessa funzione. In questo caso la funzione sarà contenuta nei diversi segmenti dei programmi, che la utilizzeranno, creando così uno spreco di spazio in memoria.

Sarebbe infatti molto meglio se la stessa funzione fosse presente in memoria in un unico segmento di codice da dove potesse essere chiamata da tutti quei programmi che intendessero utilizzarla.

Allo scopo Amiga dispone di un sistema di librerie Run-time nelle quali sono contenute diverse funzioni utilizzabili, anche contemporaneamente, da più programmi. Tutte queste funzioni sono implementate in vista di un loro utilizzo in multitask, evitando che le diverse chiamate, anche simultanee, alla stessa funzione interferiscano reciprocamente in maniera indesiderata.

### Librerie di sistema

Lo stesso sistema operativo dell'Amiga essendo costituito da diverse funzioni orientate al multitask è distribuito in un insieme di librerie Run-time dette, librerie di sistema. Queste librerie possono, a loro volta, essere distinte in librerie residenti e librerie transienti. Le prime sono quelle librerie sempre presenti nella memoria del calcolatore, in quanto memorizzate su ROM o caricate dal disco di KickStart al momento dell'accensione; mentre le seconde sono quelle librerie, generalmente presenti sul disco del WorkBench, che vengono caricate in memoria solo al momento del loro effettivo utilizzo.

Tutte le librerie principali, ovvero giudicate indispensabili per un corretto funzionamento della macchina, sono del tipo residente: (Exec.library, Gfx.library, Intuition.library, Dos.library eccetera.) mentre quelle secondarie, preposte a compiti più particolari, sono del tipo transiente: (RAM.library, Icon.library, eccetera).

### Struttura di una libreria

Come tutti i compiti correlati al multitask, anche la gestione delle librerie è affidata a Exec. Infatti a ogni libreria è associata una struttura di controllo detta Struct Library, la quale è direttamente controllata da Exec. Tale struttura è visibile in **figura 1**.

Come molte altre strutture gestite da Exec, anche la Struct Library inizia con un nodo, tramite il quale la

```
(file: include/exec/libraries.h)
struct Library
{
struct node lib_Node; /* nodo di collegamento */
UBYTE lib_Flags; /* Flags di stato della libreria */
UBYTE lib_Pad; /* Byte non utilizzato */
UWORD lib_NegSize; /* Estensione all'indietro della libreria */
UWORD lib_PosSize; /* Estensione in avanti della libreria */
UWORD lib_Version; /* Numero di versione */
UWORD lib_Revision; /* Numero di revisione */
APTR lib_IdString; /* Stringa d'identificazione */
ULONG lib_Sum; /* Somma logica di controllo */
UWORD lib_OpenCnt; /* Contatore aperture */
}
```

*Figura 1.*

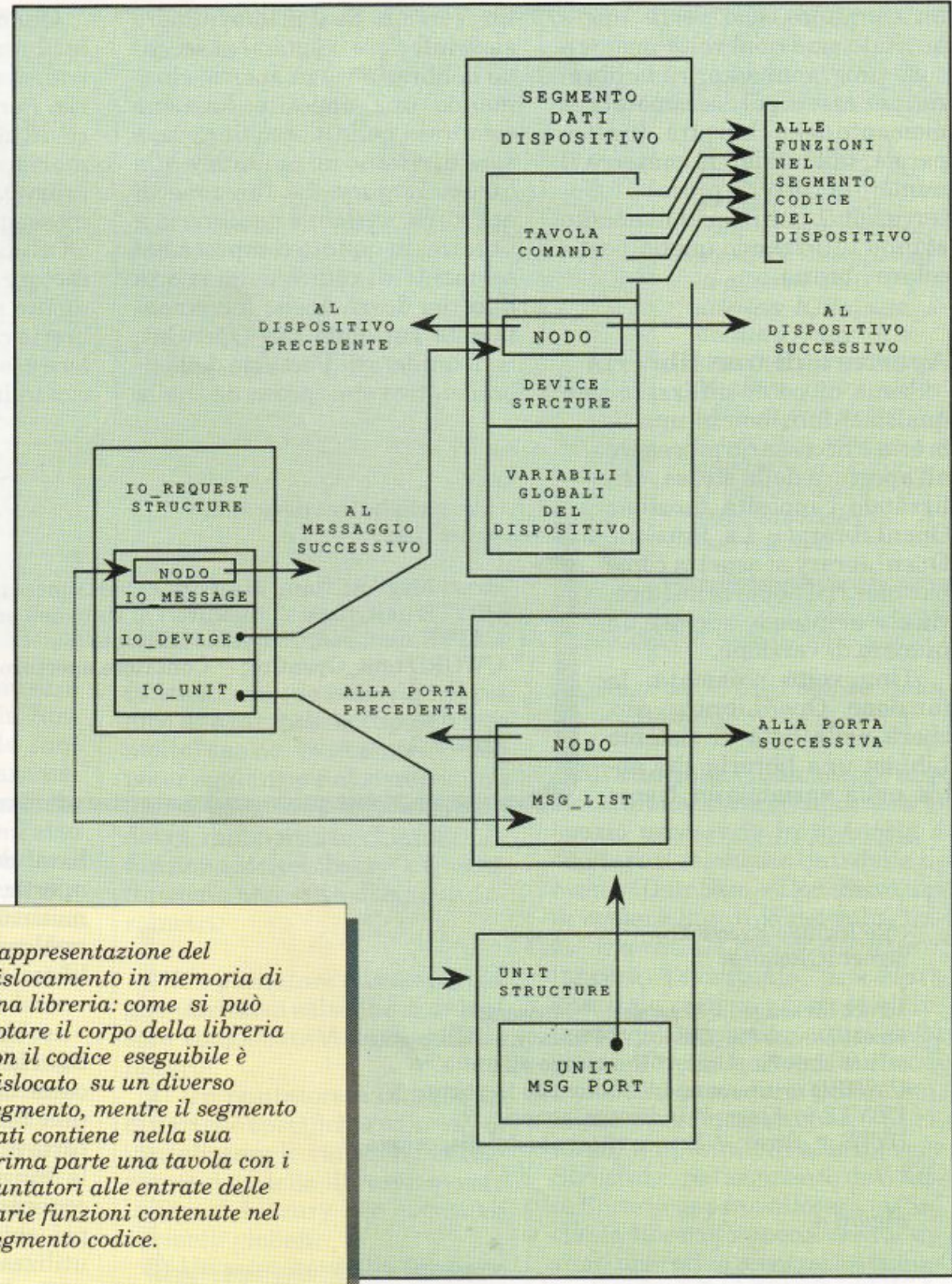
```
(file: include/exec/devices.h):
struct Device
{
struct Library dd_Device; /* Struttura di libreria utilizzata per il */
}
/* controllo del dispositivo */
```

*Figura 2.*

struttura sarà collegata alla lista di tutte le librerie presenti nel sistema. Tale lista, di nome LibList, è gestita automaticamente da Exec. La variabile Flags designa lo stato di una libreria, specificando se sono in corso operazioni di controllo della checksum, oppure la possibilità di eliminare la libreria dal sistema qualora essa non sia più necessaria.

Per ogni libreria disponibile saranno presenti in memoria un segmento di codice, con tutte le funzioni costituenti il corpo della libreria stessa, e un segmento di dati, composto da tre distinte sezioni, disposte l'una dopo l'altra. La prima sezione conterrà una tabella di istruzioni di salto che punteranno alle varie funzioni della libreria contenute nel segmento di codice.

A seguito di tale tabella sarà posizionata la struttura Library, seguita a sua volta dalle variabili globali utilizzate dalla libreria. La variabile lib_NegSize, presente nella struttura Library, designa l'estensione all'indietro del segmento dati della stessa libreria, ovvero lo spazio in memoria occupato dall'inizio di tale segmento, fino al nodo, non compreso nel compito, della struttura Library. Tale valore coincide con l'estensione in memoria della tabella di salto. La variabile lib_PosSize, indicherà invece, l'estensione in avanti del segmento dati, ovvero la memoria occupata dalla Struct Library e dalle variabili globali che la seguono.

Le variabili lib_Version e lib_Revision designeranno rispettivamente il numero di versione e di revisione della libreria. Il puntatore lib_IdString punterà invece a una stringa che generalmente contiene il nome e la data di produzione della libreria. La variabile lib_Sum conterrà il valore della cheksum del segmento dati della libreria, usato a scopo di controllo. Infine, la variabile lib_OpenCnt implementa



*Rappresentazione del dislocamento in memoria di una libreria: come si può notare il corpo della libreria con il codice eseguibile è dislocato su un diverso segmento, mentre il segmento dati contiene nella sua prima parte una tavola con i puntatori alle entrate delle varie funzioni contenute nel segmento codice.*

un contatore, che verrà incrementato ogni qual volta un qualsiasi programma aprirà la libreria, per essere poi decremetata al momento della chiusura. In altre parole, tale variabile conterrà il numero dei diversi processi (o interruzioni), che in quel momento stanno utilizzando quella particolare libreria.

## Apertura di una libreria

Prima di poter utilizzare una qualsiasi funzione in una libreria è necessario procedere all'apertura della stessa, utilizzando l'apposita funzione OpenLibrary(). La funzione OpenLibrary si aspetta come parametri il nome della libreria che si intende aprire e un numero di versione.

Una volta chiamata, la funzione OpenLibrary cercherà nella lista di sistema LibList una libreria che abbia nella variabile ln_Name della struttura lib_Node contenuta in testa alla Struct Library, un puntatore alla stringa specificata come primo parametro, nella funzione OpenLibrary(). Inoltre la stessa funzione confronterà il numero di versione, specificato come secondo parametro, con il contenuto della variabile lib_Version. Se il primo numero sarà inferiore o uguale al secondo la libreria verrà aperta, chiamando una apposita funzione compresa nella stessa libreria, e sarà di ritorno un puntatore alla Struct Library. La funzione di apertura, variabile da libreria a libreria, in quanto compresa nel segmento di codice della stessa libreria, dovrà anche incrementare la variabile Lib_OpenCnt. Se il numero di Versione della libreria trovata, presente nella variabile lib_Version, dovesse essere inferiore a quello specificato nella funzione OpenLibrary(), la libreria non verrà aperta, in quanto giudicata obsoleta, e la funzione farà ritorno alla procedura chiamante, con un parametro di ritorno nullo, a testimonianza della mancata apertura.

```
(file: include/exec/devices.h):
struct Unit
{
struct MsgPort *unit_MsgPort; /* Porta messaggi dell'unità */
UBYTE unit_flags; /* Indicatori di stato dell'unità */
UBYTE uint_pad;  /* Byte inutilizzato */
UWORD unit_OpenCnt; /* Contatore aperture unità */
}
```

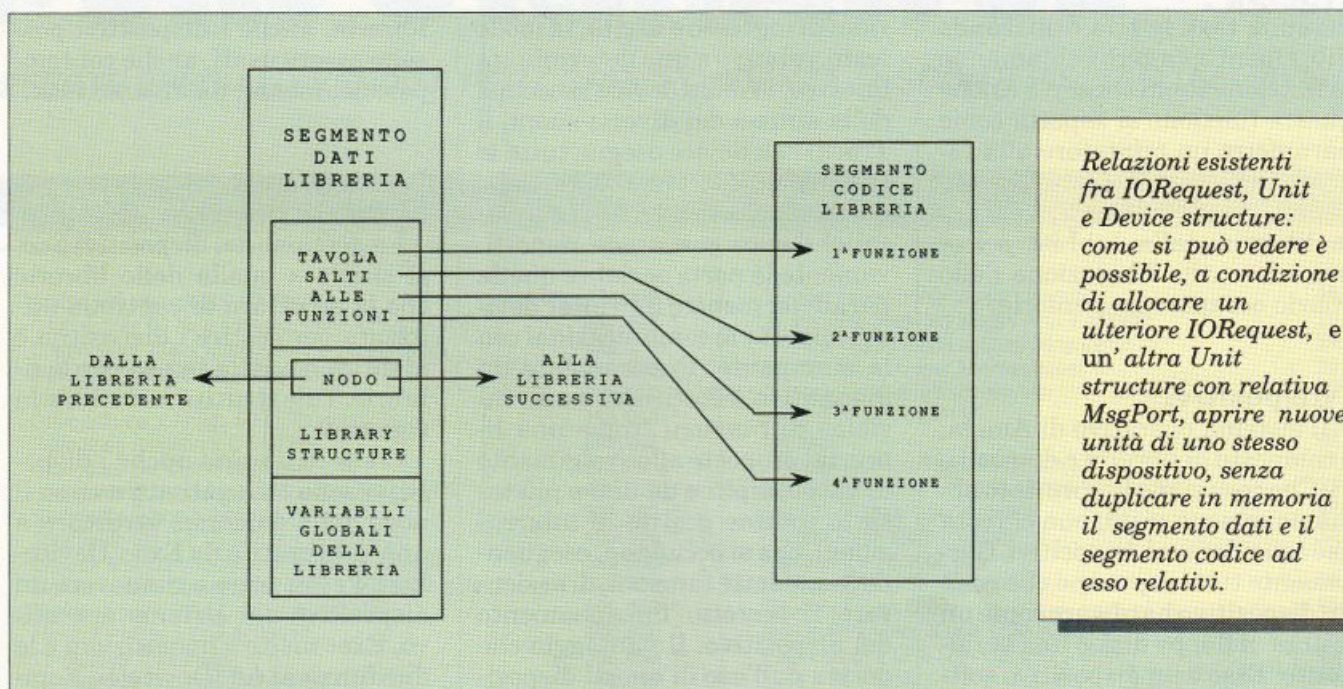
*Figura 3.*

```
(file:Include/Exec/IO.h):
struct IORequest
{
struct Message io_Message; /* Messaggio */
struct Device *io_Device; /* Puntatore al dispositivo */
struct Unit *io_Unit; /* Puntatore all'unità */
UWORD io_Command; /* Comando impartito */
UBYTE io_Flags; /* Indicatori generici */
BYTE io_Error; /* Errore ritornato dal dispositivo */
}
```

*Figura 4.*

Quanto detto, si applica sia per le librerie residenti sia per quelle transienti, già presenti in memoria. Per quanto riguarda le librerie transienti non caricate in memoria, la funzione OpenLibrary(), utilizzando il sistema dei messaggi, chiederà alla libreria DOS di cercare, prima nella directory corrente, e successivamente nella directory Libs, la libreria che si intende aprire. Qualora essa venga trovata, verrà caricata in memoria, aggiunta alla lista delle librerie (LibList) e aperta, numero di versione permettendo, con una procedura analoga a quella utilizzata per le librerie residenti.

• **Uso delle funzioni di libreria:** per poter utilizzare qualcuna delle funzioni presenti in una libreria, occorrerà innanzitutto aprire la libreria, mediante la funzione OpenLibrary(), precedentemente esaminata, per poi chiamare le varie funzioni utilizzando la tavola di salti presente in testa al segmento dati della libreria. Poiché il parametro ritornato dalla funzione OpenLibrary è un puntatore alla Struct Library si dovrà utilizzare una forma di indirizzamento indiretto con offset negativo.

• **Chiusura di una libreria:** una volta che una task abbia terminato l'utilizzo di una libreria,

*Relazioni esistenti fra IORequest, Unit e Device structure: come si può vedere è possibile, a condizione di allocare un ulteriore IORequest, e un' altra Unit structure con relativa MsgPort, aprire nuove unità di uno stesso dispositivo, senza duplicare in memoria il segmento dati e il segmento codice ad esso relativi.*

dovrà procedere alla chiusura della stessa, chiamando la funzione CloseLibrary(). Tale funzione necessiterà come parametro d'ingresso dell'indirizzo della Struct Library (fornito all'atto dell'apertura dalla funzione OpenLibrary(), e provvederà alla chiamata di un'apposita funzione di chiusura, compresa fra quelle della libreria, la quale provvederà, fra l'altro, al decremento della variabile lib_OpenCnt. Quando tale variabile raggiungerà il valore zero, indicando il termine dell'utilizzo della libreria, la Struct Library verrà scollegata dalla LibList e verrà liberato lo spazio in memoria occupato dalla libreria.

## Librerie Run-Time Utente

Alle librerie di sistema è anche possibile aggiungere delle proprie librerie, preparate dall'utente, le quali verranno inserite nella lista delle librerie disponibili. È consigliabile procedere alla creazione di simili librerie solo quando più task debbano condividere l'uso di determinate funzioni, che, se inserite nel codice di ogni task, occuperebbero molto spazio in memoria. Per poter aggiungere al sistema una nuova libreria la task creatrice dovrà innanzitutto chiamare la funzione MakeLibrary(). Questa funzione necessita di cinque parametri:

**1)** Un puntatore alla tavola degli indirizzi di salto che dovrà essere conglobata nel corpo della libreria.

**2)** Un puntatore a una tavola di dati che specifichi i valori a cui inizializzare le variabili del segmento dati della libreria, ovvero la Struct Library e le eventuali variabili globali.

**3)** Un puntatore alla funzione di inizializzazione della libreria, che verrà chiamata poco prima di aggiungere la libreria al sistema.

**4)** L'estensione della tavola dei salti.

**5)** Lo spazio occupato dalla Struct Library e dalle eventuali variabili globali della libreria. Questi ultimi due valori saranno quelli riportati nelle variabili lib_NegSize e lib_PosSize della Struct Library.

All'atto della chiamata, la funzione MakeLibrary() allocherà lo spazio necessario in memoria a ospitare il segmento dati della libreria (DataSize + CodeSize), indi copierà la tavola dei salti, inizializzerà la Library Structure, lancerà l'eventuale procedura d'inizializzazione presente all'interno del segmento codici della stessa libreria e ritornerà il controllo alla procedura chiamante, fornendo, come parametro di ritorno, un puntatore alla Library Structure, nel segmento dati della libreria appena allocato. Affinché la libreria appena creata diventi operativa occorrerà aggiungerla alla LibList utilizzando la funzione AddLibrary() la quale necessita, come unico parametro, il puntatore alla Library Structure fornito in uscita dalla funzione MakeLibrary().

## Rimozione di una libreria

Si può procedere alla rimozione di una libreria, a condizione che

nessuna task la stia utilizzando (lib_OpenCnt a zero), chiamando la funzione RemLibrary(). Anche questa funzione si aspetta come parametro un puntatore alla Library Structure. La funzione appena vista si limiterà a separare la libreria dalla LibList senza procedere alla liberazione dello spazio occupato in memoria.

## I dispositivi

Il sistema operativo di Amiga, è composto di librerie e dispositivi. Che cosa siano le librerie lo abbiamo appena visto, non ci resta che considerare i dispositivi. Certamente tutti sapranno che cos'è un dispositivo hardware come un lettore di floppy disk o una stampante. Esso è un dispositivo software, ovvero l'emulazione software del funzionamento di un dispositivo hardware.

Tutti i dispositivi presenti attualmente nell'Amiga si occupano della gestione di operazioni d'ingresso e uscita: il Keyboard device gestisce tutte le informazioni provenienti dalla tastiera, il GamePort device controlla i joystick e il mouse, l'Input device raggruppa tutte le informazioni provenienti dai vari ingressi, il Console device gestisce le operazioni di ingresso e uscita, in modo testo, relative a una determinata finestra, l'Audio device si occupa della sintesi dei diversi suoni, il TrackDisk device esegue tutte le operazioni relative alla gestione dei dischi, il Serial device e il Parallel device gestiscono rispettivamente le porta seriale e quella parallela, mentre il Printer device supporta le comunicazioni con la stampante. Praticamente un dispositivo è costituito da un insieme di funzioni (come una libreria) preposte allo svolgimento di vari compiti e da una o più task (o routine gestite in interruzione), che si occupano, eseguendo le suddette funzioni, di assicurare il corretto funzionamento del dispositivo. Il vantaggio che deriva dall'uso di questi dispositivi è notevole: infatti le diverse task per poter eseguire le loro operazioni di Input, Output non dovranno chiamare delle apposite funzioni, ma dovranno, spedire dei comandi ai dispositivi proposti a quelle particolari operazioni di Input e Output, i quali eseguiranno le istruzioni loro impartite, con dei processi paralleli, per poi rispedire indietro le eventuali risposte con lo stesso messaggio che aveva comunicato loro il comando. Analogamente alle librerie, anche i dispositivi possono essere usati, anche contemporaneamente, da diverse task.

## La struttura di un dispositivo

La gestione dei dispositivi è così simile a quella delle librerie che la struttura di controllo utilizzata per gestire i dispositivi è identica a quella in uso per le librerie. Tale struttura è visibile in **figura 2.**

Come le librerie anche i dispositivi sono collegati attraverso il nodo della omonima struttura a una lista gestita da Exec: DeviceList. Per inserire o rimuovere un dispositivo dal sistema operativo, Exec mette a disposizioni e le due funzioni AddDevice() e RemDevice(), che necessitano entrambe di un puntatore alla Device structure del dispositivo che si intende aggiungere o inserire.

A differenza di una libreria, un dispositivo può gestire diverse unità fisiche. Ad esempio, sebbene esista un unico TrackDisk device, il sistema operativo potrà gestire sino a 4unità di questo dispositivo, una per ogni driver fisico, interno o esterno, connesso al calcolatore. Ognuna di queste unità sarà controllata da un'apposita struttura detta Unit Structure come in **figura 3**.

## Apertura e chiusura di un dispositivo

Come per le librerie, prima di poter inviare un qualsiasi comando a un dispositivo occorrerà aprirlo utilizzando l'apposita funzione OpenDevice().

Questa funzione necessita dei seguenti parametri:

**1)** Un puntatore al nome del dispositivo che si intende utilizzare.
**2)** Il numero dell'unità.

```
struct IOStdReq
{
struct Message io_Message; /* Messaggio *
struct Device *io_Device; /* Puntatore al dispositivo *
struct Unit *io_Unit; /* Puntatore all'unità */
UWORD io_Command; /* Comando impartito */
UBYTE io_Flags; /* Indicatori generici */
BYTE io_Error; /* Errore ritornato dal dispositivo */
ULONG io_Actual; /* Quantità dati ancora da trasmettere o ricevere - */
ULONG io_Length; /* Quantità iniziale dati da trasmettere o ricevere */
APTR io_Data; /* Puntatore al buffer per i dati */
ULONG io_Offset;
}
```

*Figura 5.*

3) Un puntatore alla struttura IORequest, affinché venga inizializzata.
4) Infine una variabile detta flags, che conterrà ulteriori informazioni relative a quel dispositivo.

Il processo di apertura di un dispositivo da parte della funzione OpenDevice() è del tutto simile a quello di apertura di una libreria. Se il dispositivo che si intende aprire non è presente nella lista di sistema DeviceList, verrà cercato un dispositivo con un simile nome, prima nella directory corrente, successivamente nella directory devs:. Una volta trovato, il dispositivo verrà inizializzato, verranno incrementate la variabile lib_OpenCnt nella Device structure e la variabile unit_OpenCnt presente nella Unit structure preposta al controllo dell'unità che si intende utilizzare. Infine la Device structure sarà collegata alla DeviceList affinché anche altre task possano utilizzare quel dispositivo.

## Comunicazioni con un dispositivo

Per poter impartire i vari comandi a un dispositivo il sistema operativo di Amiga utilizza una particolare struttura detta IORequest che riproduciamo in **figura 4**.

Se il dispositivo di IO è un dispositivo standard, al posto della precedente struttura si utilizza la IOStdReq come si può vedere in **figura 5**.

Tali strutture dovranno essere allocate dalla task che si desideri utilizzare con un certo dispositivo e dovranno essere inizializzate dal sistema operativo mediante la funzione OpenDevice() precedentemente esaminata. Detta funzione oltre a provvedere all'apertura del dispositivo imposterà al loro giusto valore i vari membri della struttura: la sottostruttura io_Message verrà inizializzata come un qualsiasi messaggio; le variabili io_Device e io_Unit conterranno rispettivamente dei puntatori alla Device structure e alla Unit structure del dispositivo e della particolare unità che verrà utilizzata. Le rimanenti variabili della IOStdReq dovranno, invece, essere inizializzate dalla task utente. Successivamente la task che ha aperto il dispositivo, che chiameremo task utente, dovrà inserire nella variabile io_Command, un codice relativo al comando che intende impartire al dispositivo e inoltrare tale comando attraverso una di queste funzioni: DoIO() e SendIO(), che necessitano come parametro di un puntatore alla IORequest structure (o IOStdReq). Le due funzioni si distinguono in quanto la prima permette l'esecuzione delle operazioni di Input/Output in modo sincrono, mentre la seconda esegue tali operazioni con una procedura asincrona. Per l'esattezza, la funzione DoIO() metterà in stato di attesa la task chiamante, sino a quando il dispositivo non avrà terminato l'esecuzione del comando impartitogli, mentre la funzione SendIO() spedirà il comando al dispositivo senza interrompere l'elaborazione in corso nella task utente.

Se si utilizza una procedura di comunicazione asincrona(chiamando SendIO()) le funzioni CheckIO() e WaitIO(), permetteranno rispettivamente di testare il termine dell'operazione di Input/Output da parte del dispositivo e di porre la task utente in stato di attesa per tale evento. Anche le due ultime funzioni necessitano come parametro di un puntatore alla IORequest structure. Eseguito il comando di Input/Output la variabile io_Error della Request structure conterrà il codice di una eventuale condizione di errore, mentre il campo io_Actual della IOstdReq conterrà la lunghezza della stringa dati letta o scritta.

## Chiusura di un dispositivo

Alla fine dell'utilizzo del dispositivo occorrerà chiuderlo utilizzando la funzione CloseDevice(), che necessita come parametro di un puntatore alla IORequest structure utilizzata per comunicare con quel dispositivo.

**Sergio Fiorentini**

# Generazione del contratto di locazione

*Il programma comparso sul numero scorso consentiva di calcolare con facilità l'importo di qualsiasi affitto abitativo. Con questo numero vi viene data la possibilità di stampare la versione definitiva di un qualsiasi contratto di locazione ad uso abitativo e a norma di legge.*

Questo programma permette di definire un adatto formulario per la creazione, tramite stampante, di contratti di locazione ad uso abitativo, secondo quanto

prevede la legge numero 392 del 1978 (unitamente alle modificazioni a questa apportate negli anni seguenti).

I documenti che si possono ottenere sono direttamente utilizzabili dalle parti che lo devono sottoscrivere: locatore e conduttore. Legalmente il supporto cartaceo è una scrittura privata, avente validità di accordo formale tra un soggetto che concede un immobile ad uso esclusivamente abitativo (casa, appartamento, villa) e un altro soggetto che lo prende in affitto, adeguandosi a una nutrita serie di condizioni esplicitamente indicate.

Equo Canone II è adatto a ricevere in Input valori e specifiche eventualmente già trattati con il software EquoCanone1, che permette di calcolare l'ammontare base annuo in lire dell'equo canone che il conduttore deve corrispondere al locatore.

Una volta ottenuta la stampa delle copie di contratto necessarie (solitamente ne occorrono tre, in quanto una serve per l'Ufficio del registro) è suffi-

ciente che locatore e conduttore, dopo aver preso visione dei contenuti, le sottoscrivano con doppia firma leggibile.

Equo Canone II è un utile strumento di lavoro, ideale non solo per geometri, stimatori, ingegneri e agenzie immobiliari, ma anche per qualsiasi persona interessata ad affittare un immobile senza dover interpellare professionisti o comprare contratti-tipo precompilati spesso troppo generici per i singoli casi.

All'utente viene semplicemente chiesto di specificare o indicare, tramite i tasti numerici o di funzione, alcuni dati informativi di obbligatoria citazione (lo richiede la legge), tra cui: luogo e data di redazione del contratto; generalità del locatore, del conduttore e dell'immobile da locare; durata dell'accordo; canone annuo stabilito; oneri di spesa accessori previsti.

Su questi input il computer lavora velocemente, definendo volta per volta un formulario ad hoc e trasmettendo alla stampante (quella di riferimento è la Commodore MPS 1200) il messaggio software per la produzione del documento cartaceo, il quale avviene su normali fogli formato A4 (questi diventano sempre almeno tre).

È importante ricordare che il programma ha validità per i soli contratti di locazione a uso abitativo, e sono quindi esclusi tutti i diversi (ad esempio le locazioni di fabbricati industriali o di negozi), anche perché è la legge stessa, a cui si fa riferimento, che prevede le distinzioni delle varie tipologie di destinazione immobiliare.

Tramite un unico paginone grafico multicolor si può controllare con facilità lo svolgimento di tutte le routine di elaborazione, con monitoraggio effettuato da un multidisplay che permette una lettura istantanea e soprattutto facilità di interpretazione: il tutto senza possibilità di errori.

La carica sul Commodore 64 si effettua con i consueti comandi di loading, ovvero:

LOAD""

oppure:

LOAD"EQUO CANONE 2",

o anche col pratico metodo di pressione contemporanea dei tasti Commodore e Run/Stop, che causa tra l'altro l'autostart immediato a fine carica, evitando la necessità del Run attivatore.

In memoria RAM, il programma occupa uno spazio di quasi 17 Kb, che aumentano di circa 0,9 Kb a lancio avvenuto.

## Funzionamento

Terminata l'operazione di caricamento, trascorsi pochi secondi da quando il programma viene fatto partire (tramite un Run di lancio), compare la pagina monitor a fondo e bordo di colore grigio, che rimane invariata nella sua parte grafica e strutturale durante tutte le fasi di esecuzione delle routine.

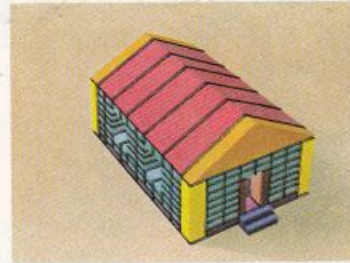
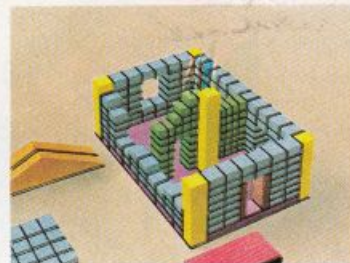
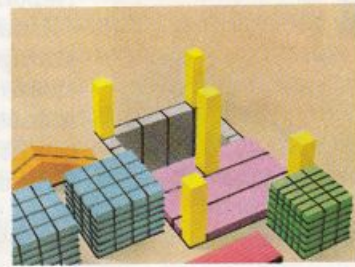
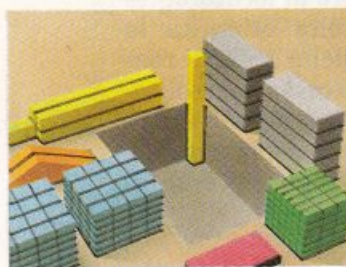
Nella zona superiore dello schermo sono presenti la grande scritta Equo Canone II (nei colori giallo-blu e bianco-rosso) e il data set di copyright (in colore grigio scuro reverse).

Un sottotitolo in blu reverse evidenzia la funzione del software, che è quella di permettere un'immediata definizione dei formulari, adatti a qualsiasi contratto di locazione ad uso abitativo, con stampa dei relativi documenti.

Nella parte inferiore dello schermo, evidenziato da una cornice marrone, trova sistemazione il multidisplay per la scrittura dei messaggi-guida, per gli input e per la proiezione della videata finale, in standby, di stampa.

Contenuti e riferimenti legislativi del programma, nonché procedure di input ed elaborazione dei dati, fanno riferimento alle disposizioni del Codice civile (in particolare agli articoli 1456, 1584, 1609 e 1756), alla legge detta dell'equo canone, numero 392 del 27/07/1978 (articolo 10 in particolare), e successive modificazioni, nonché alla vastissima legislazione vigente in materia di diritto privato, e agli usi e consuetudini locali sulle locazioni.

Anche se non esplicitamente precisato su video, si intende che l'uso del programma Equo Canone II non comporta, comunque e da chiunque venga usato, alcuna responsabilità per chi lo ha realizzato, prodotto e pubblicato: occorrerà, pertanto, controllare sempre attentamente ogni ope-

razione che porta poi alla stampa del contratto.

La scritta iniziale lampeggiante in vari colori avverte che la routine esecutiva degli Input può essere avviata solo premendo il tasto Return: in caso contrario il computer rimane in Standby.

Si inizia con la richiesta dei dati di obbligatoria citazione (a garanzia di regolarità formale del contratto).

Nell'ordine occorre specificare luogo e data di stesura del documento, per poi indicare lo status giuridico del locatore (privato, ditta o società), unitamente all'input delle relative complete generalità, che nel caso di privati sono cognome e nome, codice fiscale, località di residenza, indirizzo e numero dell'indirizzo. Se il locatore è invece una ditta, in luogo di cognome e nome va specificata la denominazione completa; così è anche per le società, e in quest'ultimo caso vanno anzi aggiunte le precisazioni di numero d'iscrizione nel registro del Tribunale locale e di cognome e nome del legale rappresentante.

Anche per l'altra parte contrattuale, cioè il conduttore (colui che riceve in affitto la casa), va ripetuto l'iter di immissione dei dati identificativi: valgono tutte le considerazioni fatte per il locatore. Segue la richiesta di Input relativamente all'immobile oggetto del contratto e nell'ordine si devono specificare il numero dei vani, la località d'indirizzo, l'indirizzo, il numero dell'indirizzo, eventuali specifiche descrittive (come scala, piano, interno, se occorre vederli poi stampati).

Altre indicazioni sono necessarie relativamente alla durata in anni del contratto, alle date d'inizio e fine locazione, all'importo annuale del canone precedentemente calcolato (magari col programma Equo Canone 1 comparso sul numero scorso), e agli oneri accessori di spesa preventivati

dal locatore.

Ultima indicazione da fornire è quella relativa al sistema di riscaldamento eventualmente presente (centralizzato oppure autonomo): nel caso di impianto comune, sul contratto dev'essere inserita una clausola particolare che vincola il conduttore ad accettare la gestione condominiale del riscaldamento, nonché a rimborsare il locatore delle relative spese per manutenzione e consumo. Con Equo Canone II la specifica del paragrafo aggiuntivo è automatica, se necessaria.

Durante le fasi di immissione dei dati, un'apposita maschera di input fornisce, di volta in volta, tutte le informazioni necessarie, affinché l'utente dia risposte adeguate: in verde chiaro, rosso e giallo viene stampato il testo della richiesta, e più sotto due indicatori di colore bianco delimitano un campo in cui viene poi visualizzato l'inserimento, che può avere un'ampiezza massima di ben 30 caratteri, se alfanumerico, oppure di 11 caratteri se puramente numerico. I dati del primo tipo prevedono l'uso dei tasti alfabetici (dalla A alla Z), numerici (dallo 0 al 9), grammaticali e semigrafici (punti, virgole, segni e barre, molto utili ad esempio per scrivere bene gli indirizzi): la visualizzazione su video avviene in colore nero, da destra verso sinistra. Per gli Input in cifre (ad esempio l'importo in lire del canone annuale) sono attivati i soli tasti numerici (dallo 0 al 9), per ottenere valori da un minimo di 0 (zero) a un massimo teorico di 999'999'999 (novecentonovantanovemilioni novecentonovantanovemila e novecentonovantanove) ogni volta. Il numero viene visualizzato in nero, già ricomposto a gruppi di tre cifre per una migliore leggibilità, da destra verso sinistra (come per i display delle calcolatrici); è possibile addirittura comporre improbabili valori frazionali premendo il tasto del punto decimale (.).

La conferma per ogni inserimento va data con Return, mentre eventuali correzioni sono sempre possibili, prima della conferma stessa, utilizzando il tasto della freccetta sinistra (sul Commodore 64 è posto di fianco a quello numerico 1).

Premendo subito e solo il tasto Return alle richieste di dati numerici, si passa oltre: l'Input sa-

rà allora considerato come inesistente, nullo, e non potrà comparire su contratto. Per altri tipi di dati (alfanumerici), l'inserimento non può invece avvenire se non si specifica qualcosa.

Per l'immissione di certe indicazioni, il multidisplay prevede una sequenza prefissata di opzioni, comodamente esplorabile a libro, in avanti e in ritorno, tramite i tasti-funzione F1 ed F3; la conferma della scelta operata va poi data con l'usuale Return. Ad esempio, quando occorre definire gli status giuridici del locatore e del conduttore, si ha una sequenza di tre opzioni: soggetto privato, oppure ditta, oppure società.

Al completamento d'esecuzione della routine di immissione dei dati viene proiettata la videata finale, un messaggio che invita l'utente ad aspettare affinché la stampante collegata al computer provveda alla stesura del contratto di locazione definito.

È possibile scegliere i fogli singoli in formato standard A4 (da inserire in sequenza) oppure i moduli continui, anche se ovviamente la prima soluzione è quella da preferire. Normalmente i contratti sono fogli a più facciate o pagine. Per ottenere la scrittura in fronte/retro basta reinserire nella stampante un foglio appena scritto ed espulso dal rullo, avendo ovviamente cura di capovolgerlo senza girarlo.

La lunghezza di un contratto-tipo, con la spaziatura interlineare prevista (10 millimetri, come d'uso in materia di scritture formali e legali) diventa sempre di almeno tre facciate, ovvero due fogli se si opera in fronte/retro (vedi esempi di stampa). In questo caso andranno uniti con colla o punti metallici controllando la corretta impaginazione del documento ottenuto.

Normalmente ogni contratto di locazione dev'essere prodotto in tre copie: una per il locatore, una per il conduttore e un'altra per l'Ufficio del registro. Il programma ne stampa una sola, per cui si può ripetere tre volte la procedura di definizione oppure, meglio, si può ricorrere all'aiuto di una fotocopiatrice: la prima pagina di ogni contratto prevede già un'apposita riga (sotto il titolo di apertura in reverse) in cui specificare a chi è destinata la copia del documento.

Ogni paragrafo della scrittura è evidenziato da un punto nero (oltre che da un numero di riferimento), mentre il testo è stampato in NLQ (near letter quality) per un'ottima leggibilità.

La composizione dei contenuti è automaticamente giustificata ai margini, al fine di conferire all'elaborato un'impaginazione veramente professionale e gradevole all'occhio di chi è tenuto a leggere la lunga sequenza di specifiche.

Il contratto assume validità a ogni effetto quando viene sottoscritto dalle due parti, mediante firma leggibile da apporre nelle apposite righe già previste alla fine dei paragrafi e (ultima pagina) dopo la clausola finale.

Ovviamente locatore e conduttore devono prendere visione del documento, prima di sottoscriverlo e di presentarne una copia all'Ufficio del registro; trattandosi di una scrittura privata, può essere cambiato (riscritto) senza difficoltà ogni volta che intervenga necessità (ad esempio per rinnovo contrattuale che preveda varianti).

La stampante di riferimento che garantisce i migliori risultati di impaginazione è la Commodore MPS 1200: altre printer possono produrre documenti di diverso lay-out per quanto riguarda grafica, caratteri e composizione, anche se ovviamente il contenuto rimane sostanzialmente invariato.

Equo Canone II ritorna automaticamente alla videata iniziale di attesa non appena termina la routine di stampa dei documenti di volta in volta definiti.

Concludendo, si può affermare che questo programma aiuta molto l'utente, effettuando tutte le procedure più noiose e complesse per la stesura dei contratti di locazione ad uso abitativo.

## Ecco gli articoli

Gli articoli del Codice civile a cui si fa esplicito riferimento nel testo da sottoscrivere sono i seguenti, di cui si specifica, per corretta informazione delle parti, il contenuto:

• **Art. 1456** (clausola risolutiva espressa). I contraenti possono convenire espressamente che il contratto si risolva nel caso che una determinata obbligazione non sia adempiuta secondo le modalità stabilite. In questo caso, la risoluzione si verifica di diritto quando la parte interessata dichiara all'altra che intende valersi della clausola risolutiva.

• **Art. 1584** (diritti del conduttore in caso di riparazioni). Se l'esecuzione delle riparazioni si protrae per oltre un sesto della durata della locazione e, in ogni caso, per oltre venti giorni, il conduttore ha diritto a una riduzione del corrispettivo proporzionata all'intera durata delle riparazioni stesse e all'entità del mancato godimento. Indipendentemente dalla sua durata, se l'esecuzione delle riparazioni rende inabitabile quella parte della casa che è necessaria per l'alloggio del conduttore e della sua famiglia, il conduttore può ottenere, secondo le circostanze, lo scioglimento del contratto.

• **Art. 1609** (piccole riparazioni a carico dell'inquilino). Le ri-

parazioni di piccola manutenzione, che a norma dell'articolo 1576 devono essere eseguite dall'inquilino a sue spese, sono quelle dipendenti dai deterioramenti prodotti dall'uso, e non quelle dipendenti da vetustà o da caso fortuito. Le suddette riparazioni, in mancanza di patto, sono determinate dagli usi locali.

• **Art. 1756** (rimborso delle spese). Salvo patti o usi contrari, il mediatore ha diritto al rimborso delle spese nei confronti della persona per incarico della quale sono state eseguite, anche se l'affare non è stato concluso.

Nel testo contrattuale si fa riferimento anche a un articolo della legge numero 392 del 27.07.1978, relativa all'equo canone, che è l'articolo seguente.

• **Art. 10** (partecipazione del conduttore all'assemblea dei condomini). Il conduttore ha diritto di voto, in luogo del proprietario dell'appartamento locatogli, nelle delibere dell'assemblea condominiale relative alle spese e alle modalità di gestione dei servizi di riscaldamento e di condizionamento d'aria. Egli inoltre ha diritto di intervenire, senza diritto di voto, sulle delibere relative alla modifica degli altri servizi comuni. La disciplina del primo comma si applica anche qualora si tratti di edificio non in condominio. In tale ipotesi i conduttori si riuniscono in apposita assemblea convocati dal proprietario dell'edificio o da almeno tre conduttori. Si osservano, in quanto applicabili, le disposizioni del Codice civile sull'assemblea dei condomini.

## Due esempi

Sono di seguito riportati due esempi di definizione contrattuale per locazioni ad uso abitativo, anche al fine di mostrare come sia facile ottenere via software risultati immediati e completi.

• **Il primo caso** tratta di un immobile sito in Milano, Piazza San Babila 21, avente 3 vani e posto in un edificio nella scala A, al secondo piano, interno 4. Viene concesso in locazione da privato a privato, e cioè dal signor Massimiliano Rossi, abitante a Viareggio (LU) in Piazza Dante 34/C, codice fiscale RSSMSS63L28F257S, al signor Arturo Brambilla residente a Monza (MI) in Viale Caduti sul Lavoro 875, codice fiscale BRMRTR35H12F339S.

Il contratto deve avere una durata di 4 anni, con inizio il 30 settembre 1989 e termine il 30 settembre 1993. L'importo annuale del canone è stato stabilito in lire 7 milioni 543mila 820 lire, più un preventivo spese accessorie di 2 milioni.

Non avendo l'immobile riscaldamento autonomo, deve comparire in contratto quanto contenuto nel relativo paragrafo di stampa numero 7.

L'accordo viene sottoscritto dalle parti a Milano il giorno 23 settembre 1989.

• **Il secondo caso** tratta invece di una casa di Aosta, indirizzo Piazza Costituzione 9/H, avente complessivamente 13 vani. Viene concessa in locazione da una società a un privato, e cioè dalla Fotoset S.p.A. (iscritta al Tribunale con numero 225'331 e legalmente rappresentata dal signor Davide Verdi), con sede sociale in Torino, Corso Cavour 44, codice fiscale 0143242435629951, alla signora Francesca Dondi residente a Saint Vincent (AO) in Via dei Pini 37, codice fiscale DNDFRN22L20H312I.

Il contratto deve avere una durata di 8 anni, con inizio il 1° giugno 1989 e termine il 1° giugno 1997. L'importo annuale del canone è stato stabilito in lire 12 milioni 114mila 790 lire, più un preventivo spese accessorie di lire 887mila.

Non avendo l'immobile riscaldamento centralizzato, non deve comparire in documento quanto contenuto nel paragrafo di stampa numero 7 del contratto relativo al precedente caso.

L'accordo viene sottoscritto dalle parti a Roma il giorno 2 gennaio 1989.

**Daniele Malavasi**

# Sei tips...più uno

*Sono ben sette i tips di questo mese e sono uno più incredibile dell'altro. Ben cinque routine per creare effetti grafici assolutamente inediti, una utility per avere contemporaneamente hires e testo sullo schermo e un tool per effettuare il dump selettivo delle variabili Basic*

L'alta risoluzione è forse uno degli strumenti meno usati da chi programma in Basic senza utilizzare espansioni. Il motivo è molto semplice: realizzare anche una semplice figura richiede lunghe e complicatissime serie di peek e poke. Un modo per ovviare all'inconveniente sarebbe quello di disporre di una finestra in alta risoluzione all'interno dello schermo normale su cui poter fare degli esperimenti, e quindi imparare a gestire correttamente l'alta risoluzione. Disponendo poi della possibilità di avere contemporaneamente hires e testo si potrebbero creare effetti interessantissimi. Per esempio, sarebbe possibile creare delle intestazioni molto spettacolari oppure realizzare una interfaccia grafica Amiga-like.

Hires and Text è la soluzione del problema; si tratta di una routine che consente di attivare una finestra in alta risoluzione sullo schermo in modo testo. Le dimensioni della finestra in alta risoluzione, possono essere controllate facilmente e cambiate in ogni momento con una semplice Sys. Per utilizzare Hires and Text dovete copiare il listato e dare il solito Run. Per attivare la finestra in alta risoluzione dovete usare questa istruzione:

sys 49152, y1, y2.

I due parametri rappresentano la prima e l'ultima linea della finestra. Per esempio, con Sys 49152, 1, 10 le prime 10 linee dello schermo (le linee sono quelle del modo testo), si trasformeranno in alta risoluzione. Tenete presente che la pagina grafica è memorizzata fra gli indirizzi 8192 e 16384 e quindi, se dovete usare la routine in un programma Basic abbastanza a lungo, vi conviene spostare l'inizio del Basic dopo l'ultima locazione destinata alla pagina grafica.

## Variable Dump

Il successo del Basic sta nella sua comodità. Pur non essendo

*Listato1. Hires and text*

```
1 rem ——————————
2 rem -                    -
3 rem -  hires and text    -
4 rem -                    -
5 rem - sys 49152,y1,y2    -
6 rem -                    -
7 rem - sys 49264 disattiva -
8 rem -                    -
9 rem ——————————
10 :
100 fort=49152 to 49324
110 reada:s=s+a:poket,a
120 next
130 if s<>20151 then print"<CLEAR>errore nei data":end
140 data 32,253,174,32,158,183,134,250,32,253,174,32,158,183,134,251,32
150 data 140,192,120,169,51,141,20,3,169,192,141,21,3,173,60,3,141
160 data 18,208,173,17,208,41,127,141,17,208,169,129,141,26,208,88,96
170 data 173,25,208,141,25,208,48,7,173,13,220,88,76,49,234,173,18
180 data 208,205,61,3,176,19,169,56,141,17,208,169,24,141,24,208,173
190 data 61,3,141,18,208,76,129,234,169,27,141,17,208,169,21,141,24
200 data 208,173,60,3,141,18,208,76,129,234,120,169,0,141,26,208,169
210 data 49,141,20,3,169,234,141,21,3,169,27,141,17,208,169,21,141
220 data 24,208,88,96,165,250,32,159,192,141,60,3,230,251,165,251,32
230 data 159,192,141,61,3,96,170,169,50,202,240,6,24,105,8,76,162
240 data 192,96,8
```

un linguaggio particolarmente potente (anzi, da questo punto di vista è di gran lunga inferiore ad altri linguaggi ad alto livello come il C e il Pascal), il Basic ha dalla sua una notevole semplicità d'uso, dovuta al fatto che si tratta di un linguaggio interpretato e non compilato.

Sviluppare software con un linguaggio interpretato come il Basic, nonostante i suoi limiti, è molto più facile che non con i linguaggi compilati. Questo naturalmente è ancora più vero se si deve usare una macchina come il C64, per il quale anche il più veloce dei compilatori allunga in modo quasi proibitivo i tempi di codifica.

L'unico neo del Basic è la mancanza di uno strumento di debugging. Provare a correggere un programma Basic per tentativi è possibile, ma non è per nulla efficiente. La cosa più importante che si deve poter osservare durante l'esecuzione del programma è il contenuto delle variabili. Si può quindi verificare la correttezza di un programma Basic avendo a disposizione uno strumento che consenta, appunto, di osservare le variabili durante l'esecuzione del programma. E questo è proprio quello che Variable Dump vi permetterà di fare.

Questa routine è davvero molto preziosa e ben presto usandola vi renderete conto che non potrete più farne a meno. Variable Dump è in grado di effettuare il Dump, anche selettivo, delle variabili usate nei programmi Basic. Il Dump è selettivo perché è possibile scegliere il tipo delle variabili da listare.

Per usare Variable Dump dovete caricarlo dalla cassetta allegata alla rivista e dare il consue-

*Listato 2. Variable Dump*

```
10 rem ―――――――
11 rem -               -
12 rem - variable dump -
13 rem -               -
15 rem ―――――――
16 :
17 t=0:k=0:fort=0to743:readd:poke49152+t,d:k=k+d:next
20 ifk<>87188 then print"<CLEAR>errore nei data":end
100 data 160,0,177,122,240,38,32,253
110 data 174,133,247,32,19,177,144,3
120 data 76,84,193,32,115,0,165,247
130 data 201,36,208,6,169,1,133,2
140 data 208,14,201,37,208,6,169,2
150 data 133,2,208,4,169,0,133,2
160 data 165,45,164,46,133,20,132,21
170 data 196,48,208,2,197,47,144,1
180 data 96,105,2,144,1,200,133,34
190 data 132,35,169,13,32,210,255,32
200 data 148,192,32,179,192,32,236,192
210 data 165,211,201,3,240,8,169,32
220 data 32,210,255,76,88,192,169,61
230 data 32,210,255,165,9,201,1,208
240 data 6,32,48,193,76,132,192,201
250 data 2,208,6,32,23,193,76,132
260 data 192,32,39,193,165,20,164,21
270 data 24,105,7,144,167,200,76,52
280 data 192,76,8,175,32,135,234,165
290 data 203,201,57,240,4,32,44,168
300 data 96,32,135,234,165,203,201,64
310 data 240,247,201,57,240,243,169,0
320 data 133,198,96,160,0,132,9,177
330 data 20,16,10,169,2,133,9,197
340 data 2,240,30,208,18,200,177,20
350 data 16,13,169,1,133,9,197,2
360 data 240,15,165,2,208,12,96,165
370 data 2,201,0,240,4,201,2,208
380 data 1,96,169,145,32,210,255,104
390 data 104,76,132,192,160,0,177,20
400 data 41,127,32,210,255,200,177,20
410 data 240,5,41,127,32,210,255,165
420 data 9,201,1,208,8,169,36,32
430 data 210,255,76,22,193,201,2,208
440 data 5,169,37,32,210,255,96,160
450 data 0,177,34,170,200,177,34,168
460 data 138,32,149,179,76,42,193,32
470 data 166,187,32,221,189,76,30,171
480 data 32,79,193,160,2,177,34,133
490 data 37,136,177,34,133,36,136,177
500 data 34,133,38,240,10,177,36,32
510 data 210,255,200,196,38,208,246,169
520 data 34,76,210,255,169,0,133,247
530 data 133,248,32,121,0,133,247,32
540 data 115,0,240,10,144,8,32,121
550 data 0,32,19,177,144,9,133,248
560 data 240,54,32,115,0,240,49,133
570 data 249,32,115,0,165,249,201,36
580 data 208,13,169,1,133,9,169,128
590 data 5,248,133,248,76,172,193,201
600 data 37,240,3,76,8,175,169,2
610 data 133,9,169,128,5,247,133,247
620 data 169,128,5,248,133,248,208,4
630 data 169,0,133,9,165,47,164,48
640 data 133,20,132,21,196,50,208,7
650 data 165,20,197,49,208,1,96,160
660 data 0,152,170,177,20,197,247,208
670 data 7,200,177,20,197,248,240,40
680 data 160,2,177,20,24,101,20,133
690 data 249,144,1,232,200,177,20,24
700 data 101,21,133,21,138,240,2,230
710 data 21,165,249,133,20,197,49,208
720 data 206,165,21,197,50,208,200,96
730 data 160,4,177,20,133,249,170,200
740 data 177,20,153,55,3,169,0,153
750 data 127,3,200,177,20,153,55,3
760 data 169,0,153,127,3,202,208,231
770 data 200,152,24,101,20,133,251,144
780 data 1,232,165,21,133,252,138,240
790 data 2,230,252,169,141,32,210,255
800 data 32,148,192,32,236,192,169,40
810 data 32,210,255,165,249,133,250,165
820 data 250,56,233,1,10,168,185,133
830 data 3,170,185,132,3,32,205,189
840 data 198,250,165,250,240,8,169,44
850 data 32,210,255,76,63,194,169,41
860 data 32,210,255,169,61,32,210,255
870 data 165,251,133,34,165,252,133,35
880 data 165,9,201,1,208,7,32,48
890 data 193,169,3,208,16,201,2,208
900 data 7,32,23,193,169,2,208,5
910 data 32,39,193,169,5,24,101,251
920 data 133,251,144,2,230,252,166,249
930 data 138,10,56,233,2,168,140,255
940 data 3,185,133,3,24,105,1,153
950 data 133,3,185,133,3,208,9,185
960 data 132,3,24,105,1,153,132,3
970 data 185,132,3,217,60,3,208,8
980 data 185,133,3,217,61,3,240,3
990 data 76,43,194,202,240,18,169,0
1000 data 153,132,3,153,133,3,136,136
1010 data 169,141,32,210,255,76,161,194
```

*Listato 3. Special effects*

```
1 rem ───────────────
2 rem -                            -
3 rem - Effetto 1                     -
4 rem -                            -
5 rem - poke 49166,x:poke49184,x+2 -
6 rem -                            -
7 rem ───────────────
8 :
10 fori=49152to49204:reada:pokei,a:next:sys49152:end
20 data 120,173,32,208,72,173,17,208,41,239,141,17,208,162,0,142,32,208,173,1
30 data 220,201,255,208,13,232,234,234,234,234,234,224,2,208,236,76,13,192,173
40 data 17,208,9,16,141,17,208,104,141,32,208,88,96,208

1 rem ───────────────
2 rem -                            -
3 rem - Effetto 2                     -
4 rem -                            -
5 rem - poke 49166,x:poke49184,x+2 -
6 rem -                            -
7 rem ───────────────
8 :
10 fort=49152 to 49203:readd:poket,d:next:sys 49152
20 data 120,173,32,208,72,173,17,208,41,239,141,17,208,162,1,142,32,208,173,1
30 data 220,201,255,208,12,232,234,234,234,234,224,3,208,237,76,13,192,173,17
40 data 208,9,16,141,17,208,104,141,32,208,88,96,96

1 rem ───────────────
2 rem -                            -
3 rem - Effetto 2                     -
4 rem -                            -
5 rem - poke 49166,x:poke49183,x+2 -
6 rem -                            -
7 rem ───────────────
8 :
10 fort=49152 to 49203:readd:poket,d:next:sys 49152
20 data 120,173,32,208,72,173,17,208,41,239,141,17,208,162,1,142,32,208,173,1
30 data 220,201,255,208,12,232,234,234,234,234,224,3,208,237,76,13,192,173,17
40 data 208,9,16,141,17,208,104,141,32,208,88,96,96

1 rem ───────────────
2 rem -                            -
3 rem - Effetto 4                     -
4 rem -                            -
5 rem - poke 49166,x:poke49179,x+3 -
6 rem -                            -
7 rem ───────────────
8 :
10 fort=49152 to 49199:readd:poket,d:next:sys49152
20 data 120,173,32,208,72,173,17,208,41,239,141,17,208,162,1,142,32,208,173,1
30 data 220,201,255,208,8,232,224,4,208,241,76,13,192,173,17,208,9,16,141,17
40 data 208,104,141,32,208,88,96,32

1 rem ───────────────
2 rem -                            -
3 rem - Effetto 5                     -
4 rem -                            -
5 rem - poke 49166,x:poke49179,x+3 -
6 rem -                            -
7 rem ───────────────
8 :
10 fort=49152 to 49199:readd:poket,d:next:sys49152
20 data 120,173,32,208,72,173,17,208,41,239,141,17,208,162,1,142,32,208,173,1
30 data 220,201,255,208,8,232,224,3,208,241,76,13,192,173,17,208,9,16,141,17
40 data 208,104,141,32,208,88,96,32
```

to Run. Abbiamo ritenuto opportuno evitarvi la fatica di digitare il listato di Variable Dump data la grande quantità di istruzioni DATA in esso presenti. Per effettuare il Dump delle variabili di un programma dovete usare queste istruzioni:

• **Sys 49152, $:** per listare tutte le variabili stringa.

• **Sys 49152, %:** per listare tutte le variabili intere.

• **Sys 49152, c:** per listare il contenuto del vettore di nome C. Per esempio: con Sys 49152, a, viene listato il contenuto dell'array di reali a(); con sys 49152, a$ viene listato l'array di stringhe a$(); con Sys 49152, a% viene listato l'array di interi a%().

• **Sys 49152:** lista tutte le variabili del programma ad eccezione degli array.

## Special effects

Gli effetti speciali sono sempre utili, e non solo per realizzare presentazioni a effetto, ma anche per rendere più piacevoli le attese in un programma serio, o semplicemente per richiamare l'attenzione dell'utente. Con questa mini raccolta di effetti grafici, veramente spettacolari, potrete cambiare look ai vostri programmi e renderli molto più accattivanti. Le cinque routine proposte hanno la stessa struttura: si caricano tutte in 49152 e si attivano tutte automaticamente quando vengono lanciate. Per interrompere lo Show dovete premere la barra di spazio.

Potete modificare l'effetto grafico realizzato da ogni routine "pokando" dei valori qualsiasi nelle locazioni indicate nelle prime linee Rem di ogni routine. Nei listati viene indicata la relazione che deve sussistere fra i valori da pokare, ma può anche non essere rispettata. Anzi, vi conviene provare tutte le possibili combinazioni di valori per scoprire l'effetto che vi piace di più. Tutte le routine possono essere inserite nei vostri programmi piuttosto facilmente. Non occupano infatti la Ram Basic, inoltre, quando premete lo spazio per interrompere l'effetto, il controllo ritorna all'interprete.

**Daniele Maggio**

# Le funzioni dell'editor

*Oltre agli ultimi comandi messi a disposizione dall'interprete, in questa puntata si parlerà dell'editor, il potente strumento integrato nell'ambiente di sviluppo che vi permetterà di scrivere i programmi velocemente e senza sforzo.*

**Terza puntata**

Nella puntata precedente si è parlato soprattutto dell'interprete di comandi, o Shell. I comandi descritti non esauriscono assolutamente l'argomento, sia perché di comandi ce ne sono ancora, sia perché esiste la possibilità di strutturare i comandi disponibili per formare vere e proprie procedure di Shell.

In questa puntata, oltre a terminare la descrizione dell'interprete, vedremo nei dettagli come funziona l'editor integrato nell'ambiente di sviluppo. Questo potentissimo strumento vi permetterà di scrivere i programmi molto più facilmente e velocemente di quanto siate mai riusciti a fare con l'editor del Basic.

E ora iniziamo con i comandi dell'interprete.

## Gli ultimi comandi dell'interprete

• **Edit.**

Questo comando serve per invocare l'editor a pieno schermo di Qpl. Normalmente l'editor è residente in memoria, e quindi viene attivato immediatamente. Volendo però potete anche liberare lo spazio occupato dall'editor (usando l'opzione B del compilatore, come sarà spiegato in futuro). In questo caso prima di invocare l'editor dovete assicurarvi che questo sia sul disco nel drive. Opzionalmente nel comando si può indicare un parametro che rappresenta il nome del file che si vuole editare. Potete usare il nome di un file esistente, o il nome di un file che non avete ancora creato. In quest'ultimo caso l'editor vi avvertirà che state per editare un file nuovo. L'editor verrà descritto più avanti in questa stessa puntata.

• **Files.**

Visualizza il contenuto del disco nel drive. In pratica questo comando equivale a: load"$",8 + list. È possibile filtrare i file da listare proprio come in Basic utilizzando i metacaratteri (?) e (*). Ecco qualche esempio:

- file pr* : *lista tutti i file che iniziano con pr.*
- file 1: *lista tutti i file del disco nel drive 1.*
- file >p: *stampa i nomi di tutti i file (verrà stampato un solo nome per linea).*

• **Fill.**

Riempie una zona di memoria con un determinato valore. In

pratica questo comando equivale a un ciclo di poke del Basic. La sintassi del comando è:Fill inizio fine val.

I primi due parametri sono gli indirizzi, in notazione esadecimale, di inizio e fine della zona di memoria. Il parametro val è il valore con cui deve essere riempita l'area e può essere un valore esadecimale o un carattere ASCII. Ecco qualche esempio:

- Fill 4800 4923 0 *riempie l'area di memoria compresa fra gli indirizzi esadecimali $4800 e $4923 con il valore $0.*
- Fill 4800 4923 'g' *riempie la stessa area di memoria con il codice ASCII del carattere g.*

(Il carattere va sempre racchiuso negli apici singoli). Prima di usare questo comando assicuratevi che l'area che volete riempire sia effettivamente utilizzabile. In caso contrario rischiereste di causare il crash del sistema. La **tavola 1** mostra la mappa di memoria al momento del lancio di Qpl. Dovete sempre fare riferimento a essa per stabilire se una zona di memoria è disponibile o meno.

• **Fkey.**
Visualizza le stringhe associate ai tasti funzione e permette anche di modificare tali associazioni. La sintassi del comando è: fkey [nt str].

I due parametri sono opzionali, e se non sono indicati, l'effetto del comando è di listare le stringhe associate ai tasti funzione. Il parametro nt rappresenta il numero del tasto funzione e str la stringa che si vuole associare. Nella stringa possono essere inclusi soltanto i caratteri visualizzabili. Se volete inserire degli spazi dovete racchiudere l'intera stringa tra doppi apici. Ecco qualche esempio:

- Fkey *lista le stringhe associate ai tasti funzione*
- Fkey 1 pippo *associa al tasto funzione F1 la stringa pippo*
- Fkey 2 "uno due tre" *associa al tasto funzione 2 la stringa indicata tra doppi apici (i doppi apici non sono inclusi ma servono soltanto per indicare i limiti della stringa).*

• **Get.**
Carica in memoria un programma Qpl compilato, o una routine in linguaggio macchina senza eseguirli. Al termine del caricamento verrà visualizzata la mappa di memoria, che mostrerà la posizione del programma. A questo punto potrete eseguire il programma, digitandone il nome, nel caso si tratti di un programma Qpl, oppure usando il comando Go, nel caso si tratti di un programma in linguaggio macchina.

Se il programma indicato è in linguaggio macchina verrà rilocato, a meno che il suo nome non venga inserito fra doppi apici. In questo caso, il programma sarà caricato nell'area di memoria da cui è stato salvato.

Per caricare un programma Qpl compilato non c'è bisogno di indicare l'estensione .C perché essa viene aggiunta automaticamente.

• **Help.**
Visualizza l'elenco dei comandi principali dell'interprete indicandone sommariamente anche la sintassi.

• **Job.**
Esegue in modo Batch un file contenente dei comandi, detto anche job. Un file di comandi trattato in questo modo è da considerare come un nuovo comando. Per editare il file dovete usare l'editor di Qpl e scrivere tutti i comandi su linee separate. Al termine dovete salvare il file mettendo nel nome l'estensione .j.

Esiste un file Batch speciale, bootscript.j che se si trova sul disco quando lanciate Qpl viene automaticamente eseguito dall'interprete (un po' come il file autoexec in Ms-Dos).

Avete anche la possibilità di passare dei parametri al file di comandi. Il meccanismo utilizzato per questo scopo è quello dei parametri posizionali. Ogni parametro è identificato da un numero che indica la sua posizione sulla stringa di comando.

Vediamo un esempio per chiarire meglio la situazione. Supponiamo di aver editato il seguente job:

```
edit £1.s
delete £1.L
```

## Tavola 1. Mappa di memoria di Qpl

*Lo spazio a disposizione dell'utente è quello indicato accanto alle voci free space e free workspace.*

| | | |
|---|---|---|
| object programs | $4f00- | (0) |
| free space | $4f00-$98ff | (18944) |
| active workspace | $9900- | (0) |
| free workspace | $9900-$a0ff | (2048) |
| shared variables | $a100- | (0) |
| exec./edit space | $a200-$cfff | (11776) |
| total space | $4f00-$cfff | (33024) |

```
compile £1 l=£1.L
unload £1
£1 testdata.d
```

Ora supponiamo di usare il seguente comando:

```
job do.j mioprg
```

Questo comando determinerà l'esecuzione dei comandi:

```
edit mioprg.s
delete mioprg.l
compile mioprg l=mioprg.l
unload mioprg
mioprg testdata.d
```

Ogni occorrenza di £1 viene automaticamente sostituita con mioprg, appunto il primo parametro passato dopo do.j.

Nel job potete anche inserire dei commenti facendoli precedere da un punto e virgola ( ;).

**• Macro.**

Definisce una macro. Ecco un esempio che dovrebbe chiarire il funzionamento del comando. Digitate il seguente comando:

```
macro uno, due; tre.
```

D'ora in poi, l'interprete sostituirà a ogni occorrenza della stringa £1 con la stringa uno, due, tre. Il meccanismo è lo stesso che regola il comportamento del comando job. Per esempio, digitando il comando:

```
macro uno, due, tre,
```

l'interprete d'ora in poi sostituirà a ogni occorrenza delle stringhe £1, £2 e £3 rispettivamente con le stringhe uno, due e tre.

**• Map.**

Visualizza la mappa della memoria fornendo la quantità di memoria disponibile per i programmi, lo spazio occupato dai programmi e gli indirizzi di inizio e fine delle aree occupate da ciascuno di questi. La **tavola 1** contiene l'output prodotto dal comando subito dopo che è stato attivato Qpl. Di seguito viene riportata la spiegazione di ogni termine:

- *Object programs:* indica lo spazio occupato da tutti i programmi in memoria (eccetto la memoria occupata dalle variabili globali).
- *Free space:* indica lo spazio ancora disponibile per i programmi e le variabili. Questo spazio può essere usato per incrementare le dimensioni del workspace (con il comando Ws). Le dimensioni di quest'area possono essere aumentate deallocando dei programmi (con Unload), riducendo le dimensioni del workspace o deallocando l'editor.
- *Active workspace:* indica le dimensioni della parte di workspace usata.
- *Free workspace:* indica le dimensioni della parte inutilizzata del workspace. Il valore visualizzato può essere modificato con il comando Ws.
- *Shared variables:* indica le dimensioni dello spazio allocato per le variabili globali non inizializzate (vettori e variabili reali).
- *Exec. / Edit space:* indica le dimensioni dello spazio occupato dall'editor e dall'interprete.
- *Total space:* indica le dimensioni massime dello spazio disponibile per i programmi, se si deallocano tutti i programmi e l'editor e si pone a zero il workspace.

**• Noreal.**

Elimina dal supporto runtime le routine per la gestione dei nu-

**Tavola 2. I comandi dell'interprete**

| Comando | Descrizione |
|---|---|
| **Color** | fissa i colori di visualizzazione e di fondo |
| **Copy** | copia un file |
| **Cs** | cancella lo schermo |
| **Date** | cambia la data |
| **Delete** | cancella un file |
| **Diskcm** | invia un comando al drive |
| **Dump** | visualizza il contenuto della memoria |
| **Dyno** | attiva/ disattiva il velocizzatore del disco |
| **Edit** | attiva l'editor |
| **Files** | visualizza il contenuto del disco |
| **Fill** | riempie la memoria con un valore |
| **Fkey** | visualizza le stringhe associate ai tasti funzione |
| **Get** | carica in memoria un programma senza eseguirlo |
| **Go** | manda in esecuzione una routine lm |
| **Help** | visualizza i principali comandi built in |
| **Job** | esegue un job |
| **Macro** | definisce una macro |
| **Map** | visualizza la mappa della memoria |
| **Noreal** | disattiva il modulo runtime per i numeri reali |
| **Pause** | attiva un ciclo di attesa |
| **Quit** | disattiva il Qpl |
| **Rename** | cambia il nome a un file |
| **Set** | carica un valore in una locazione |
| **Size** | visualizza le dimensioni di un file |
| **Type** | visualizza il contenuto di un file |
| **Unload** | dealloca un file dalla memoria |
| **Ws** | fissa le dimensioni del workspace |

meri reali. Usando questo comando si guadagnano circa 2.5 Kb di memoria extra per i programmi.

Tenete presente che l'esecuzione di questo comando comporta la cancellazione della memoria (cioè tutti i programmi in memoria vengono persi), e che per poter usare nuovamente i numeri reali nei vostri programmi dovrete lanciare nuovamente Qpl.

**• Pause.**
Inserisce una pausa in un job. La sintassi del comando è: pause [str]. Il parametro str, che è opzionale, indica il messaggio da visualizzare durante la pausa. Per far terminare il ciclo d'attesa si deve premere Return.

Questo comando è molto utile se usato nei job perché permette di inserire dei messaggi e delle pause durante l'esecuzione dei comandi.

**• Quit.**
Resetta il C64 e quindi disattiva Qpl.

**• Rename.**
Permette di cambiare nome a un file. La sintassi è: rename vecchionome nuovonome.

**• Set.**
Carica una locazione di memoria con un particolare valore. In pratica set equivale all'istruzione Poke del Basic. La sintassi di questa istruzione è set ind val, dove ind è l'indirizzo della locazione, e val è il valore da memorizzare. Entrambi i valori devono essere espressi in notazione esadecimale.

**• Size.**
Visualizza le dimensioni di un programma in Qpl senza caricarlo in memoria. La sintassi del comando è size nomefile, dove nomefile è il nome del file di cui si vuole conoscere le dimensioni.

**• Type.**
Visualizza il contenuto di un file testo o di un programma sorgente Qpl. Questo comando non può essere usato per vedere il contenuto di un programma compilato. Ecco alcuni esempi d'uso:

- type pippo.s *visualizza il contenuto del file pippo.s (l'estensione .s è obbligatoria perché , se non viene indicata, l'interprete considera l'estensione di default,.c.)*
- type pippo.s >p *stampa il contenuto del file pippo.s.*
- type l *visualizza la libreria standard.*

Potete rallentare la visualizzazione del file premendo il tasto Ctrl, o bloccarla temporaneamente tenendo premuto il tasto Run/Stop.

**• Unload.**
Dealloca un programma. La sintassi del comando è: unload [nomeprg]. Il parametro nomeprg, opzionale, indica il nome del programma da rimuovere dalla memoria. Se non viene indicato nessun parametro vengono rimossi tutti i programmi dalla memoria. Se il parametro è indicato e corrisponde a un programma presente in memoria, viene rimosso insieme a tutti programmi che risiedono più in alto nella memoria.

**• Ws.**
Permette di modificare le dimensioni del workspace. Il comando ammette due formati: ws dim e ws clear. Il primo formato serve per dimensionare esattamente il workspace. Il parametro dim indica la dimensione, espressa in notazione esadecimale, del workspace. I valori minori di $28 vengono interpretati come numero di blocchi di 1K. Il secondo formato serve invece per pulire il workspace. Vediamo qualche esempio:

- ws a *riserva 10 blocchi di 1 Kb per il workspace.*
- ws 200 *riserva 512 byte per il workspace.*

Questo era l'ultimo comando messo a disposizione dall'interprete. La **tavola 2** contiene l'elenco di tutti i comandi, compresi quelli visti nella puntata precedente.

## L'editor

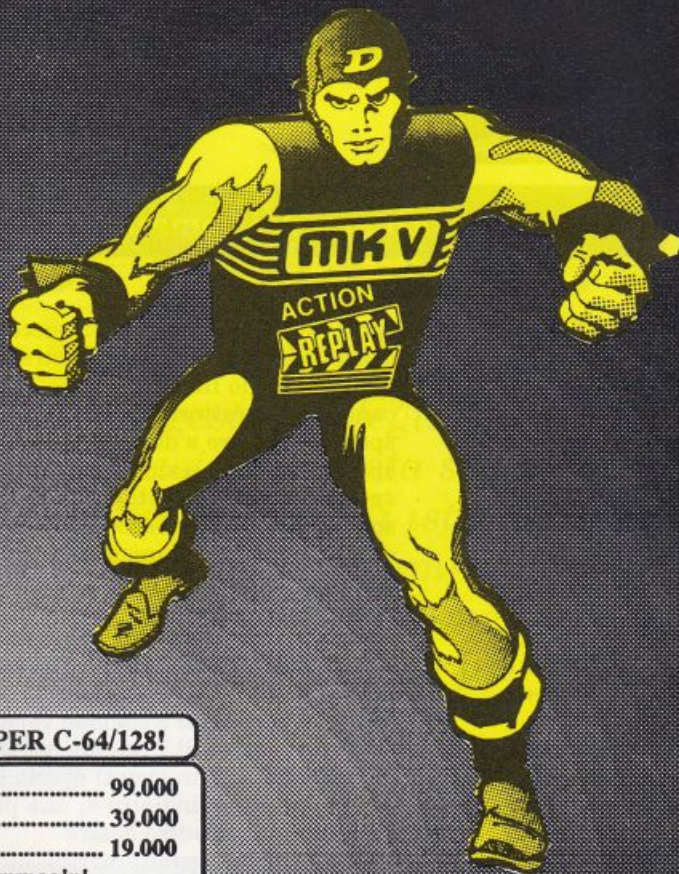
L'editor integrato nell'ambiente di sviluppo è essenziale e potente. Mette a disposizione tutti i comandi necessari per scrivere un programma in modo agevole e pur non avendo la complessità di un Word Processor è comunque molto più potente e comodo dell'editor del C64.

La lunghezza del testo, che può essere editato su 125 colonne, è limitata soltanto dalla disponibilità di memoria. La **tavola 3** contiene l'elenco e una breve descrizione dei comandi di editing disponibili. Oltre a quelli ce ne sono altri associati ai tasti funzione, e sono visualizzati nella parte bassa dello schermo durante l'inserimento del testo. Eccone la descrizione:

**• Del line.**
Cancella la linea in cui si trova il cursore. Si attiva premendo il tasto funzione F1.

**• Ins line.**
Inserisce una linea prima di quella in cui si trova il cursore. Si attiva con il tasto funzione F2.

**• Mark.**
Permette di marcare un blocco di testo. La porzione selezionata potrà poi essere elaborata con uno dei comandi descritti di seguito. Per selezionare il blocco di testo, spostate il cursore sulla prima linea del blocco (o sull'ultima), premete il tasto funzione

Mk V' non solo è la miglior cartridge per effettuare copie di sicurezza del proprio software, ma è anche il più efficace velocizzatore nastro/disco e la più versatile cartuccia di utility esistente.
La sua peculiarità più innovativa è data dal suo microprocessore interno, appositamente studiato per sovrapporsi a quello del computer, ed assolutamente invisibile al sistema.
Ad esempio, mentre le altre cartucce si fermano ad un banale "Sprite Killer" per facilitare i giochi, Mk V' è in grado di trovare automaticamente le "Poke" necessarie per le vite infinite di qualsiasi programma presente e futuro, senza attendere che siano pubblicate dalle riviste o che qualche smanettone studi il programma. Ora anche voi potete produrre giochi "trainer", senza alcuna conoscenza di linguaggio macchina! Inoltre sprotegge e porta da nastro a disco (e viceversa) qualsiasi programma protetto, anche in multiload (con i parametri in dotazione); può trasferire molti programmi e files dal formato 5"1/4 al nuovo 1581 da 3"1/2; velocizza il nastro 5-6 volte oppure 8-10 volte, con velocità selezionabile; velocizza il disk drive come se fosse parallelo (2 velocità: 202 blocchi in 9 secondi oppure in 6!), ed è sempre efficace, anche con i programmi che disabilitano i fastload normali. Mk V' incorpora un vero e proprio editor di schermo, per poter cambiare più facilmente e velocemente le scritte nelle schermate o nei programmi; funziona da interfaccia parallela, per collegare una qualsiasi stampante standard Centronics al C64/128 ed usarla all'interno di qualsiasi programma, anche grafico; stampa o di salva in qualsiasi momento la schermata o gli sprites di un gioco, per alterarli a piacimento. Aggiunge nuovi comandi al Basic, monitor L/M e disk, crea serie di immagini in sequenza su nastro, e tantissime altre cose ancora.
**Per Commodore 64 e 128 (in modo 64), con qualsiasi registratore o disk drive, originali o compatibili.**

## ASSICURATI ANCHE TU LA MIGLIOR CARTUCCIA PER C-64/128!

**Mk V', manuale in italiano, garanzia 5 anni ........ 99.000**
**Cavo Centronics per Mk V' ........ 39.000**
**Enhancement Disk - utilities e parametri speciali ........ 19.000**
**Graphic Disk, nuovo disco di utility per Mk V con SlideShow di immagini, Sprite Editor Deluxe, Message Maker ad altro ancora ........ 19.000**

### MODEM

Tutti i modem sono Hayes compatibili e comprendono il software di gestione. I modelli esterni sono dotati di alimentatore proprio e sono compatibili con tutti i computer con porta seriale RS-232; i modelli interni si inseriscono in tutti i PC e compatibili.

300/1200 interno ........ 189.000
300/1200/2400 interno ........ 319.000
300/1200 esterno ........ 239.000
300/1200/2400 esterno ........ 349.000
300/1200/2400/4800 esterno ........ 849.000
300/1200/2400/4800/9600 ........ 1.899.000
300/1200/2400 specifico per PS/2 ........ 489.000
300/1200 + Videotel interno ........ 229.000
300/1200 + Videotel esterno ........ 299.000
300/1200/2400 + Videotel esterno ........ 739.000

### DIGITALIZZATORI

Framer, il miglior digitalizzatore per Amiga, in tempo reale e con una qualità video stupefacente ........ 1.199.000
Sampler 64, professionale campionatore di suoni per C64/128, non si limita a portare nella memoria del computer i suoni ma li elabora per creare effetti incredibili , compatibile MIDI, con un potentissimo live-sequencer ed editor. Fornito con ComDrum, una batteria elettronica professionale, vi farà ricredere sulle potenzialità musicali del vostro C64 ........ 149.000
Digitalizzatore audio per Amiga stereofonico ........ 179.000
Scheda digitalizzatrice video per PC real-time ........ telefonare

### SHORT-CIRCUIT MAKER

Dispositivo per tutti i computer dotati di porta parallela, crea all'interno del vostro computer uno strano fumo azzurrognolo rendendo inservibile la macchina. Utile per convincere il capufficio o i genitori a cambiare computer con un modello più recente.
In offerta a 29.000 rimborsabili all'acquisto del computer nuovo.

### SUPER LIGHT PEN

Eccezionale penna ottica di alta qualità, sfrutta componenti elettroluminosi eccellenti per una precisione millimetrica che non si riscontra assolutamente nelle concorrenti; disegnare col computer diventa più facile e veloce, mantenendo la precisione che può dare un ottimo mouse. Il rapporto qualità-prezzo è favorevolissimo.
**Per Amiga o per PC con scheda grafica ........ 99.000**

### ROBOT-ARM

Esplora anche tu le meraviglie della robotica, con questo completo braccio automatico; dispone di ben 5 assi di movimento, per una versatilità unica! Controllato semplicemente da una coppia di joystick, oppure dal computer con l'apposita interfaccia (opzionale), che permette di creare lunghe serie di movimenti da far ripetere al braccio oltre ad altre grandi possibilità di programmazione. Completo di accessori come pinze, pale, stabilizzatori ed attacchi magnetici.

**Robot-Arm ........ 159.000**
**Interfaccia per Amiga ........ 89.000**
**Interfaccia per C64 ........ 89.000**

### PORTADISCHETTI

Nuovi modelli in ABS antiurto, antistatico, antiacido ed autoestinguente, dal design modernissimo, geniale e decisamente bello. I modelli più piccoli (portatili) hanno una apertura a ventaglio per consentire una ricerca veloce dei dischetti, ed una chiusura completamente ermetica.

3"1/2 5 pz. colorati ........ 4.000
3"1/2 10 pz. ........ 5.000
3"1/2 20 pz. ........ 15.000
3"1/2 40 pz. ........ 20.000
3"1/2 80 pz. ........ 28.000
3"1/2 Posso 150 pz. ........ 39.000
5"1/4 8 pz. colorati ........ 5.000
5"1/4 50 pz. ........ 22.000
5"1/4 100 pz. ........ 26.000
5"1/4 180 pz. Posso ........ 44.000

### COPERTINE PER COMPUTER

Indispensabili per proteggere il vostro computer,monitor e stampante da polvere, graffi, liquidi o umidità che potrebbero danneggiare i vostri investimenti ed il vostro lavoro. Realizzati su misura per decine di marche diverse, sono disponibili nelle versioni in PVC di colore grigio metallizzato, lavabile, antistatico, autoestinguente e molto robusto. In alternativa, per le sole tastiere, sono disponibili in plexiglass rigido trasparente, oppure ancora una pellicola trasparente che va stesa sopra i tasti, elegante ed innovativa, poichè permette di lavorare continuando a proteggere la tastiera.
**Diversi prezzi per i vari modelli - telefonare**

## Viale Monte Nero 31
## 20135 Milano
### Tel. (02) 55.18.04.84
### (4 linee ric. aut.)
### Fax (02) 55.18.81.05 (24 ore)

Negozio aperto al pubblico tutti i giorni dalle 10 alle 13 e dalle 15 alle 19.
Vendita per corrispondenza.
Sconti per quantità ai sigg. Rivenditori.

## I prezzi Flopperia sono IVA compresa, sempre!

## Tavola 3. Elenco dei comandi di editing dell'editor

| | |
|---|---|
| **Return** | sposta il cursore sulla linea successiva |
| **Del** | cancella il carattere alla sinistra del cursore |
| **Inst** | abilità il modo Insert |
| **Ctrl + crsr left** | cancella il carattere alla sinistra del cursore e trascina il testo a sinistra |
| **Crsr right** | sposta il cursore a destra |
| **Crsr left** | sposta il cursore a sinistra |
| **Crsr up** | sposta il cursore in alto |
| **Crsr down** | sposta il cursore in basso |
| **Home** | sposta il cursore all'inizio del file |
| **Ctrl + x** | cancella la linea in cui si trova il cursore |
| **Ctrl + k** | cancella dalla posizione del cursore fino alla fine della linea |
| **Ctrl + y** | sposta il cursore alla fine della linea |
| **Ctrl + [** | sposta il cursore all'inizio della linea |
| **Ctrl + a** | attiva/ disattiva il set maiuscolo |
| **Ctrl + i** | sposta il cursore di 2 colonne verso destra |
| **Ctrl + u** | sposta il cursore di 2 colonne verso sinistra |
| **Ctrl + n** | sposta il cursore in avanti di una pagina (1 pagina = 20 linee) |
| **Ctrl + p** | sposta il cursore indietro di una pagina |
| **Ctrl + j** | tab di 2 colonne verso destra |
| **Ctrl + o** | tab di 2 colonne verso sinistra |
| **Ctrl + w** | ridisegna lo schermo a partire dalla colonna in cui si trova il cursore |
| **Ctrl + v** | ridisegna lo schermo a a partire dalla prima colonna |
| **Ctrl + b** | richiama l'ultimo comando inserito. |

F3, spostate il cursore sull'ultima linea (o sulla prima) del blocco e premete nuovamente il tasto funzione F3. Le linee selezionate verranno poste in reverse.

**• Write.**
Scrive sul disco il blocco selezionato. Selezionato il comando con il tasto funzione F4, dovete inserire il nome del file e premere Return.

**• Find.**
Permette di cercare una stringa o una linea. Questo comando, che si attiva con il tasto funzione F5, ammette due formati: find n e find str. Il primo formato serve per spostare il cursore su una particolare linea, individuata dal numero n. Il secondo formato permette invece di cercare una stringa nel testo. La stringa deve essere indicata fra doppi apici. La ricerca della stringa parte sempre dalla posizione del cursore, e va verso la fine del file terminando sull'ultimo carattere del testo. I due formati possono essere combinati. Ecco qualche esempio:

- find "pippo" *cerca la stringa pippo a partire dalla posizione del cursore fino alla fine del file.*
- find 1 "pippo" *cerca la stringa pippo a partire dalla prima linea del testo indipendentemente dalla posizione del cursore.*
- find +30 *sposta il cursore di 30 linee in avanti.*
- find -30 *sposta il cursore di 30 linee indietro.*
- find +1 "pippo" *cerca la stringa pippo a partire dalla linea successiva rispetto a quella in cui si trova il cursore. Usando il comando Crtl+ b si può simulare la ripetizione della ricerca.*

**• Change.**
Permette di sostituire una stringa. La sintassi del comando, che si attiva con il tasto funzione F6, è: change s1 s2. La stringa s1 viene sostituita dalla stringa s2. Entrambe le stringhe vanno poste tra apici singoli. Quando viene trovata la stringa s1 appare il messaggio:

change this string (y/n/c=cancel)?

e quindi potete scegliere se sostituire la stringa e interrompere l'operazione, premendo Y, oppure proseguire nella ricerca, premendo N, oppure terminare l'operazione, premendo C.

**• Help.**
Visualizza l'elenco di tutti i comandi di editing disponibili. Si attiva con il tasto funzione F7.

**• Quit.**
Fa uscire dall'editor.

**Daniele Maggio**
*(continua)*

# 200 volte più potente

*Le istruzioni di controllo, quelle per la gestione degli errori e per l'output controllato, sono i protagonisti di questa seconda puntata dedicata a Ultra Basic. Il primo gruppo di istruzioni è quello che sicuramente appassionerà ogni programmatore: le istruzioni di controllo di Ultra Basic sono più numerose e potenti di quelle di ogni altro linguaggio disponibile per C64.*

**Seconda puntata**

Nella puntata precedente si è accennato brevemente alle potenzialità di Ultra Basic. Tuttavia, per apprezzare veramente i pregi di questa stupenda espansione, è necessario toccare con mano gli strumenti più interessanti messi disposizione.

Tra questi sicuramente ci sono le istruzioni di controllo, particolarmente potenti e assolutamente senza alcun riscontro in altri linguaggi per C64. Inizieremo la puntata proprio da queste.

*Figura 1.*

*Figura 2.*

## Istruzioni di controllo

Ultra Basic, mette a disposizione un insieme di istruzioni di controllo che è nettamente più vasto di quello disponibile in qualsiasi altro linguaggio (Pascal e C compresi) o in qualsiasi altra espansione per C64. Sfruttando adeguatamente tutte le istruzioni di questo gruppo, dovrebbe essere, estremamente più semplice e immediato sviluppare software di qualunque tipo.

Inoltre, occorre sottolineare che le istruzioni di controllo del Basic standard (If, Then, For, Next, eccetera) sono sempre disponibili, e quindi si sommano a quelle nuove di Ultra Basic. Ecco le nuove istruzioni di controllo a disposizione:

• **If - then - else:** questa istruzione di controllo è una estensione della nota if - then del Basic standard. Questa istruzione si comporta in questo modo: viene valutata l'espressione booleana, condizione, che compare dopo la parola chiave If. Se questa risulta vera, viene eseguita l'istruzione indicata dopo la parola chiave Then altrimenti, cioè se la condi-

*Figura 3/4.*

zione risulta falsa, viene eseguita l'istruzione posta dopo la parola chiave Else.

Nella **figura 1** è riportato un esempio d'uso di questa struttura di controllo. Come si può notare dall'esempio sia dopo Then che dopo Else si può mettere un solo statement; inoltre prima di Else va sempre posto il carattere (:) di due punti .

• **If - then - begin - bend:** questa struttura di controllo è una estensione della precedente. Con questa istruzione si possono mettere più statement e linee di programma nei rami Then e Else.

Il funzionamento di questa istruzione di controllo è assolutamente analogo a quello dell'istruzione precedente: viene valutata la condizione indicata dopo la parola chiave if, e se risulta vera vengono eseguite tutte le istruzioni e linee di programma che compaiono fra le parole chiave Begin e Belse.

Nel caso in cui la condizione sia falsa, vengono eseguite tutte le istruzioni e linee di programma comprese fra Belse e Bend. Nella **figura 2** viene riportato un breve esempio d'impiego di questa struttura di controllo. Dall'esempio si possono trarre alcune importanti osservazioni: le istruzioni da eseguire nei due casi possono comparire anche sulla stessa linea in cui si trovano le parole chiavi Begin e Belse e in questo caso la prima istruzione va preceduta dal carattere (:) di due punti. Il ramo Belse è del tutto opzionale; tuttavia se lo omettete dovete comunque specificare la parola chiave Bend.

```
1 REM ------------
2 REM -            -
3 REM - PROC CALL -
4 REM -            -
5 REM ------------
6 :
10 A=1:B=1:Z=0
15 CALL COLORE (A*7,B-1,Z,"PROVA")
20 END
30 PROC COLORE (D,E,F,H$)
35 CLS
40 COLOR D,E,F:CENTER H$
50 GOTO 20
```

*Figura 5.*

• **Repeat - until:** è una struttura di iterazione e funziona in questo modo: vengono eseguite tutte le istruzioni o linee di programma comprese fra le parole chiave Repeat e Until. Quindi viene valutata la condizione indicata dopo la parola chiave Until e se questa risulta falsa vengono nuovamente eseguite tutte le istruzioni, mentre se è vera l'esecuzione del programma prosegue con l'istruzione posta dopo Until (e dopo la condizione).

La **figura 3** contiene un esempio d'uso di questa struttura di controllo.

Come si può notare dall'esempio, le istruzioni da eseguire più volte possono anche essere poste sulla stessa linea (in questo caso come separatore vanno (:) i due punti). Inoltre si possono mettere delle istruzioni dopo la condizione che compare accanto alla parola chiave Until.

• **Loop - exit [if] - end loop:** è una struttura di iterazione che non trova posto in nessun linguaggio ad alto livello, neanche nel C o nel Pascal. È molto utile anche se in realtà può essere facilmente simulata con l'aiuto delle istruzioni Goto e If - Then (però si perde molto in termini di leggibilità). Tale struttura si comporta in questo modo: vengono eseguite tutte le istruzioni comprese fra le due parole chiave Loop e End loop, e se viene incontrata la parola chiave Exit si fuoriesce dal ciclo. Dopo la parola chiave Exit si può mettere opzionalmente una condizione. In questo caso si fuoriesce dal ciclo solo se la condizione risulta vera. Se fra Loop e End loop non ci sono Exit, il ciclo verrà ripetuto all'infinito.

Un esempio d'uso di questa struttura di controllo è contenuto nella **figura 4**.

• **Proc - end proc:** permette di definire delle procedure. Piuttosto che usare le parole e meglio utilizzare un esempio per chiarire il funzionamento di questa struttura.

Prima di guardare i programmi d'esempio, riportati nelle **fi-**

gure 5 e 6, dovete dare un'occhiata alla definizione delle prossime due istruzioni.

• **Call:** permette di richiamare una procedura ed equivale a un Goto. La procedura richiamata con questa istruzione (e solo con essa) non necessariamente deve contenere la parola chiave End proc alla fine del blocco, che la definisce. Nella chiamata va specificato il nome della procedura e tutti gli eventuali parametri.

• **Exec:** permette di richiamare una procedura ed equivale all'istruzione Gosub. La procedura richiamata con questa istruzione deve necessariamente terminare con la parola chiave End proc.

• **Sgoto:** questa istruzione è una estensione del Goto tradizionale, implementa infatti il cosiddetto Goto calcolato. In questa istruzione il numero della linea a cui si vuole saltare può essere una espressione. Ecco un esempio d'uso:10 input a: sgoto 100+a*10

• **Pop:** permette di uscire da un ciclo prima che questo si concluda normalmente o di annullare l'effetto di un Gosub (più precisamente pop rimuove il contenuto della cima dello stack). La sintassi di questa istruzione è Pop x dove x può assumere uno dei seguenti valori:

**1:** per annullare l'effetto di un Gosub
**2:** per uscire da una procedura chiamata con Exec
**3:** per uscire da un ciclo Repeat - until
**4:** per uscire da un ciclo Loop - end loop

• **Reset:** permette di settare il puntatore alla linea data. Con questa istruzione in pratica potete gestire come volete la lettura delle linee, data nel programma. La sintassi dell'istruzione è: Reset l, dove l, che può essere un numero o una espressione, rappresenta la linea a cui deve fare riferimento l'istruzione Read.

• **Case of - when - otherwise - endcase:** è una delle strutture di controllo più potenti fra quelle a disposizione. È analoga al case disponibile in Pascal con l'aggiunta della clausola otherwise. La struttura si comporta in questo modo: viene valutata l'espressione indicata dopo la parola chiave Case of. Il valore ottenuto viene confrontato con la lista di valori che compare dopo ogni when e se nella lista è compreso tale valore, vengono eseguite tutte le istruzioni fino al prossimo When; l'esecuzione riprende dall'istruzione successiva a Endcase. Se il valore dell'espressione non è contenuto in nessuna delle liste indicate, vengono eseguite, se indicate, le istruzioni comprese fra Otherwise e Endcase. Un esempio d'uso di questa struttura è contenuto nella **figura 7**. È da notare che in ogni clausola while si possono specificare più valori oppure un solo valore. Inoltre le istruzioni da eseguire in ogni singolo caso, compreso Otherwise, possono comparire su più linee di programma.

*Figura 6.*

*Figura 7.*

## Gestione degli errori

Le istruzioni di questo gruppo potrebbero benissimo rientrare nella categoria supporto alla programmazione, perché anch'esse semplificano notevolmente la stesura dei programmi.

Eccone una descrizione dettagliata:

• **Help:** è una delle istruzioni che userete più spesso, e di cui alla lunga non potrete assolutamente più fare a meno. Questa

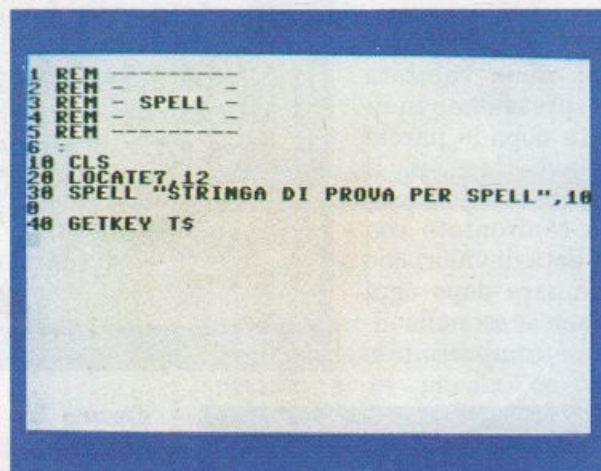
istruzione, che non necessita di alcun parametro, deve essere usata quando l'interprete segnala un errore, e visualizza la linea in cui è stato riscontrato l'errore ponendo in reverse ciò che lo ha causato.

• **On error goto:** anche questa è una istruzione utilissima. Permette di eseguire un salto a una determinata linea di programma ogni volta che l'interprete rileva un errore durante l'esecuzione del programma. La sintassi di questa istruzione è: on error goto n, dove n individua la parte di programma da eseguire non appena viene riscontrato un errore. Una volta incontrata questa istruzione l'interprete non segnalerà più gli errori ma si limiterà a eseguire la parte di programma indicata. Per disabilitare questo comportamento si usa la prossima istruzione.

• **No error:** indica all'interprete di riprendere a segnalare gli errori e quindi serve per disabilitare l'effetto dell'istruzione On error goto. Non necessita di alcun parametro.

• **Resume [next]:** questa istruzione praticamente equivale a un Return. Può essere usata solo nel pezzo di programma che viene eseguito, quando l'interprete incontra un errore e fa riprendere l'esecuzione del programma dalla linea in cui è stato riscontrato l'errore. L'istruzione ha due formati: Resume e Resume next. Usando il primo formato l'esecuzione del programma riprende dalla linea in cui è stato trovato l'errore e precisamente dall'istruzione che lo ha causato. Il secondo formato fa riprendere l'esecuzione dall'istruzione successiva a quella che ha determinato l'errore.

## Output controllato

• **Locate:** consente di portare il cursore in una posizione qualsiasi dello schermo. Può essere usata in due formati: locate x, y, o locate x, y, stringa. Nel primo caso l'istruzione si limita a posizionare il cursore nella posizione di coordinate (x, y), nel secondo caso visualizza, a partire dalla posizione selezionata, la stringa indicata come terzo parametro.

• **Center:** centra il testo sulla linea in cui questo viene visualizzato. La linea dello schermo in cui viene visualizzato il testo centrato è quella in cui si trova il cursore prima dell'esecuzione dell'istruzione.

• **Chrlo:** attiva il set minuscolo. Equivale a printchr$(14).

• **Chrhi:** attiva il set maiuscolo. Equivale a printchr$(142).

• **Cls:** pulisce lo schermo. Equivale a printchr$(147).

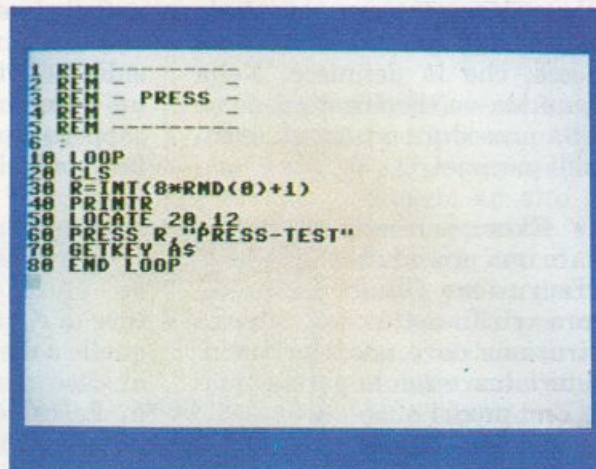



*Figura 9.*

*Figura 8.*

• **Spell:** permette di visualizzare una stringa con un effetto molto particolare.

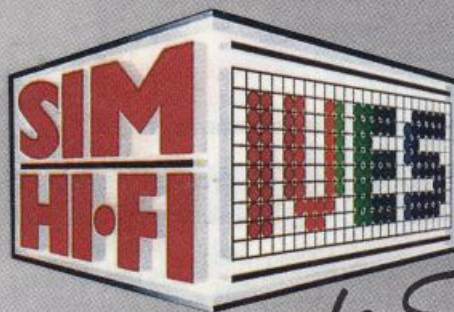
La sintassi di questa istruzione è: spell str, v. Il primo parametro è la stringa da visualizzare, che può essere anche una variabile, mentre il secondo parametro indica la velocità di visualizzazione.

In **figura 8** potete vedere un esempio d'uso di questa istruzione.

• **Press:** descrivere a parole il funzionamento di questa istruzione risulterebbe troppo lungo e servirebbe soltanto a complicare la situazione.

Per vedere che cosa fa press lanciate il programma in **figura 9**. È necessario che il parametro r, che compare nell'istruzione, sia sempre un intero compreso fra 1 e 8.

**Daniele Maggio**

*(continua)*

## Il mistero del Modo Decimale

Leggo da moltissimo tempo *Radio Elettronica & Computer* e ho imparato veramente molto sul Commodore 64, anche per quanto riguarda la programmazione in linguaggio macchina. Malgrado questo, c'è nel mio bagaglio di conoscenze su questo sistema una lacuna che non riesco a colmare. In sostanza non mi appare chiaro l'uso del Modo Decimale e l'uso di quelle istruzioni assembler direttamente interessate all'uso di numeri concepiti come decimali. Potreste gentilmente far luce su questo intricatissimo problema?

**Alan Giordani**
Luino (Va)

*Il Modo Decimale consente al 6502, microprocessore del C64, di considerare i numeri (e le operazioni tra questi) in modo decimale piuttosto che binario (o in Assembler esadecimale). Trattando numeri piccoli (minori di 100) e volendo stampare il contenuto binario di un byte in modo decimale, non superiore a 99, è sufficiente una breve routine in linguaggio macchina come quella del* ***listato 1****. In pratica occorre trasformare il numero binario (base 2) in un numero decimale (base 10) e poi fare stampare le decine e le unità dalla routine di stampa del Kernal (****tavola 1*** *e listato 1): mediante un monitor, digitate il listato 1. Al termine tornate al Basic e digitate Poke $2040,23. In questo modo, come tutti saprete, viene posto il valore 23 ($17 esadecimale e 00010111 nella locazione $2040 in forma binaria). Digitate SYS8192 per attivare la micro-ruotine del listato 1 e vedrete apparire il numero 23. Fin qui nulla di troppo complesso. Trattando numeri più grandi, però, questa operazione potrebbe necessitare di molti byte di programma. Occorrerebbe sottrarre prima le unità, le decine (come nel listato 1), le centinaia e le migliaia: artificioso e poco elegante! La via migliore per risolvere questo problema, è l'uso del Modo Decimale, settato mediante l'istruzione assembler SED. Nel Modo Decimale un numero minore di 100 viene considerato diviso in due singoli numeri inferiori a 9 (zero incluso), che costitiscono le decine e le unità. In particolare un numero binario di 8 bit rappresenta, con i primi 4 bit a sinistra le decine, con i restanti 4 le unità. La radicale differenza rispetto al Modo Binario è il diverso modo di agire delle istruzioni assembler ADC e SBC. Supponiamo di avere in memoria il numero binario 00010111 (cioè $17 esadecimale). Un programmatore, pensando in modo esadecimale (per interpretare più comodamente il formato binario), lo considererebbe come una volta 16 (che neologisticamente potremmo definire una "sedicina") sommata a 7 (le unità), cioè 23 decimale. Il microprocessore impostato in Modo Binario compirebbe lo stesso processo logico per dare l'Output decimale di quel numero binario. Un microprocessore impostato in Modo Decimale, invece, fa quello che farebbe un fanciullo di prima elementare quando gli si chiedesse di leggere la versione esadecimale del numero in questione. Il numero $17 viene considerato proprio 17 (una decina e sette unità). Se in Modo Binario spingiamo il 6502 a sommare $1 e $9, avremo come risultato $0A ,cioè 10. Nulla di più logico. Ma se compiamo la stessa operazione in Modo Decimale avremo come risultato $10, ovvero 16! Il* ***listato 2*** *risolve lo stesso problema svolto dal listato 1. Per convertire il numero, però, utilizza il Modo Decimale. Il listato 2 appare decisamente più lungo, ma questo è dovuto al fatto che l'esempio è molto semplice e il numero è piccolo.*

*Listato 1. Modo Binario*

```
2000 LDA $2040      ;fase1
2003 LDX #$00       ;fase2
2005 CMP #$0A
2007 BCC $200E      ;fase3
2009 SBC #$0A
200B INX
200C BNE $2005      ;fase4
200E TAY
200F TXA
2010 ORA #$30
2012 JSR $FFD2      ;fase5
2015 TYA
2016 ORA #$30
2018 JSR $FFD2
201B RTS
```

*Listato 2. Modo Decimale*

```
2000 LDA $2040
2003 STA #2041
2006 LDA #$00
2008 LDX #$00
200A SED            ;Modo Decimale!
200B ASL $2041      ;il Loop torna qui
200E STA $2042
2011 ADC $2042      ;raddoppia
2014 INX
2015 CPX #$08
2017 BNE $200B
2019 CLD            ;Modo Binario!
201A TAY
201B LSR            ;estrae le decine
201C LSR
201D LSR
201E LSR
201F ORA #$30
2021 JSR $FFD2
2024 TYA
2025 AND #$0F       ;prende le unità
2027 ORA #$30
2029 JSR $FFD2
202C RTS
```

**Tavola 1. Fasi del flusso della routine**

*Fase 1: contatore delle decine a 0*
*Fase 2: se il valore è minore di 10 vai a Fase 4*
*Fase 3: sottrai 10 e aggiungi 1 al contatore delle decine; vai a Fase 2*
*Fase 4: stampa il contatore come numero decimale.*
*Fase 5: stampa il resto come unità*